



TUGAS AKHIR - KI141502

RANCANG BANGUN KERANGKA KERJA SISTEM INFORMASI AKADEMIK MODULAR BERBASIS WEB DENGAN POLA ARSITEKTUR *HIERARCHICAL MODEL- VIEW-CONTROLLER* DAN *REPOSITORY SERVICE PATTERN*

FARHAN RAMADHANA
NRP 05111440000078

Dosen Pembimbing
Rizky Januar Akbar, S.Kom., M.Eng.
Dr. Ir. Siti Rochimah, M.T.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KI141502

**RANCANG BANGUN KERANGKA KERJA SISTEM
INFORMASI AKADEMIK MODULAR BERBASIS WEB
DENGAN POLA ARSITEKTUR *HIERARCHICAL
MODEL-VIEW-CONTROLLER* DAN *REPOSITORY
SERVICE PATTERN***

**FARHAN RAMADHANA
NRP 05111440000078**

**Dosen Pembimbing
Rizky Januar Akbar, S.Kom., M.Eng.
Dr. Ir. Siti Rochimah, M.T.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi Dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

**SOFTWARE DESIGN AND IMPLEMENTATION OF
MODULAR ACADEMIC INFORMATION
SYSTEM'S FRAMEWORK BASED ON WEB WITH
HIERARCHICAL MODEL-VIEW-CONTROLLER
ARCHITECTURE AND REPOSITORY SERVICE
PATTERN**

FARHAN RAMADHANA
NRP 05111440000078

Supervisor
Rizky Januar Akbar, S.Kom., M.Eng.
Dr. Ir. Siti Rochimah, M.T.

Department of Informatics
Faculty of Information and Communication Technology
Sepuluh Nopember Institut of Technology
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN KERANGKA KERJA SISTEM INFORMASI AKADEMIK MODULAR BERBASIS WEB DENGAN POLA ARSITEKTUR *HIERARCHICAL MODEL- VIEW-CONTROLLER* DAN *REPOSITORY SERVICE PATTERN*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada

Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh

Farhan Ramadhana
NRP: 05111440000078

Disetujui Oleh Dosen Pembimbing Tugas Akhir

1. Rizky Januar Akbar, S.Kom., M.Eng.
NIP: 198701032014041001

2. Dr. Ir. Siti Rochimah, M.T.
NIP: 196810021994032001



SURABAYA

Juni, 2018

[Halaman ini sengaja dikosongkan]

**RANCANG BANGUN KERANGKA KERJA SISTEM
INFORMASI AKADEMIK MODULAR BERBASIS WEB
DENGAN POLA ARSITEKTUR *HIERARCHICAL MODEL-
VIEW-CONTROLLER* DAN *REPOSITORY SERVICE
PATTERN***

Nama Mahasiswa : Farhan Ramadhana
NRP : 05111440000078
Departemen : Informatika FTIK-ITS
Dosen Pembimbing 1 : Rizky Januar Akbar, S.Kom., M.Eng.
Dosen Pembimbing 2 : Dr. Ir. Siti Rochimah, M.T.

Abstrak

Pada saat ini Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) ITS sedang melakukan perubahan Sistem Informasi Akademik (SIKAD) secara besar-besaran. Hal tersebut dilakukan karena sistem yang lama sudah tidak sesuai dengan keadaan pada saat ini. Banyak perubahan yang terjadi di kampus ITS sehingga mengakibatkan fungsionalitas yang terdapat pada SIKAD harus diubah dan diganti dengan yang baru.

Sistem Informasi Akademik (SIKAD) pada perguruan tinggi dirancang untuk memudahkan dalam pengelolaan informasi seperti pengelolaan nilai mahasiswa, pengambilan matakuliah, penjadwalan matakuliah, menampilkan indeks prestasi semester, menampilkan prestasi kumulatif dan proses-proses yang berkaitan dengan akademik lainnya yang terdapat di perguruan tinggi. Semua proses tersebut nantinya akan diterapkan ke setiap modul-modul yang berkaitan. Adapun modul-modul tersebut antara lain: kurikulum, pembelajaran, ekivalensi, SAR, penilaian, wisuda, penjadwalan, FRS, SKEM, SKPI, dll.

Adapun kekurangan dari SIAKAD yang dimiliki ITS saat ini adalah sistem yang tidak modular, sehingga menyulitkan bagi pengembang untuk melakukan pengembangan terhadap sistem dan memakan banyak waktu untuk melakukan perubahan tersebut.

Pada tugas akhir ini akan membangun kerangka kerja sistem akademik yang bersifat modular dengan menggunakan arsitektur Hierarchical Model-View-Controller dan Repository-Service-Pattern. Pada kerangka kerja SIAKAD yang akan dibangun akan menerapkan RBAC (Role Based Access Control) sebagai aturan untuk hak akses user, protokol OAuth2 sebagai otorisasi, API, internationalization, push notification, dan logging.

Kata kunci: *Sistem Informasi Akademik, OAuth2, RBAC, API, internationalization, push notification, dan logging.*

**SOFTWARE DESIGN AND IMPLEMENTATION OF
MODULAR ACADEMIC INFORMATION SYSTEM'S
FRAMEWORK BASED ON WEB WITH HIERARCHICAL
MODEL-VIEW-CONTROLLER ARCHITECTURE AND
REPOSITORY SERVICE PATETRN**

Name	: Farhan Ramadhana
NRP	: 05111440000078
Department	: Informatika FTIK-ITS
Supervisor 1	: Rizky Januar Akbar, S.Kom., M.Eng.
Supervisor 2	: Dr. Ir. Siti Rochimah, M.T.

Abstract

Nowadays, Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) ITS is doing change on the information system of academic (SIKAD) massively. It's done because of the existing system is not relevant to the present condition. Many changes that happen at ITS caused the existing functionality needs to be renewed.

SIKAD on the colleges are designed to facilitate information management such as students scores management, choosing the subjects, classes scheduling, showing student's GPA and other processes that relevant to the academic matters in the college. All those processes will be applied to every relevant module. Those modules are curriculum, learning, equivalency, SAR, scoring, graduation, scheduling, FRS, SKEM, SKPI, etc.

As for the deficiency of existing SIKAD owned by ITS is the system which is not modular, so it is difficult for the developer to develop the system and take much time to do the change.

In this thesis, author will develop the framework of modular academic system by using Architectural Hierarchical Model-View-Controller and Repository-Service-Pattern. On the SIKAD

framework, which will be built, will be applied RBAC (Role Based Access Control) as the rules for user rights access, OAuth2 protocol as authorization, API, internationalization, push notification, and logging.

Keywords: *Academic System Information, OAuth2, RBAC, API, internationalization, push notification, and logging.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah, segala puji bagi Allah S.W.T, yang atas izin dan karuniaNya penulis dapat menyelesaikan Tugas Akhir yang berjudul **“RANCANG BANGUN KERANGKA KERJA SISTEM INFORMASI AKADEMIK MODULAR BERBASIS WEB DENGAN POLA ARSITEKTUR *HIERARCHICAL MODEL-VIEW-CONTROLLER* DAN *REPOSITORY SERVICE PATTERN*”**.

Dalam pegerjaan Tugas Akhir ini, penulis mendapatkan banyak sekali ilmu baru dan memperdalam ilmu-ilmu yang sebelumnya telah diajarkan selama masa perkuliahan di Informatika ITS.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari banyak pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah S.W.T dan Nabi Muhammad S.A.W.
2. Ibu Fauziah (ibu), S.E dan bapak Prof. Dr. Muhammad Adam, S.E., MBA (ayah) selaku kedua orang tua penulis yang selalu memberi dukungan, bimbingan, doa maupun nasehat kepada penulis.
3. Aidil Syahputra S.E., MM., dr. Faradila Savitri, Raihan Ramadhani S.E, dan Jihan Fathia selaku saudara kandung penulis yang selalu menyemangati dan mendoakan penulis dari jauh selama berada di tanah rantau.
4. Bapak Rizky Januar Akbar, S.Kom., M.Eng. selaku pembimbing I yang telah membantu, membimbing, dan memberikan ilmunya kepada penulis dalam menyelesaikan Tugas Akhir ini.
5. Ibu Dr. Ir. Siti Rochimah, M.T. selaku pembimbing II yang telah membantu, membimbing, dan memberikan ilmunya kepada penulis dalam menyelesaikan Tugas Akhir ini.

6. Bapak Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Departemen Informatika ITS, Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc. selaku koordinator Tugas Akhir, dan segenap dosen Informatika ITS yang telah memberikan ilmunya.
7. Teman-teman seperjuangan Tugas Akhir di lab RPL: Valdy, Aldo, Faishal, Rara, Elva, Kurnia, Winda, Raras, Nurul, Angga, Sabila, Arya, Ikhsan, Steven, William dan lainnya serta seluruh teman-teman C1E yang telah menemani dan memberikan semangat kepada penulis selama pengerjaan Tugas Akhir.
8. Admin-admin laboratorium RPL angkatan 2014, 2015, dan 2016 yang selalu memberi dukungan terhadap penulis.
9. Teman-teman kontrakan sutorejo: Aulia, Zainal, Hidayat, Sakti, Fauzan, sahabat-sahabat penulis: Muariff, Rifqi, Nanda, Rijal, Zarwan, Kasyfil, Yusfaizi, dan teman-teman asal Aceh angkatan 2014 yang selalu setia menemani penulis selama berada di tanah rantau.
10. Teman-teman Angkatan 10 (Generate) SMAN 10 Fajar Harapan yang selalu dan setia kepada penulis sejak SMA sampai saat ini.
11. Pihak lain yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa dalam pembuatan Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga penulis meminta maaf sebesar-besarnya terhadap kesalahan-kesalahan tersebut. Penulis sangat mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2018

Penulis

DAFTAR ISI

Abstrak	vii
Abstract	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Tujuan	4
1.5 Manfaat	5
1.6 Metodologi	5
1.6.1 Penyusunan Proposal Tugas Akhir.....	5
1.6.2 Studi Literatur	6
1.6.3 Analisis dan Desain Perangkat Lunak	6
1.6.4 Implementasi Perangkat Lunak	6
1.6.5 Pengujian dan Evaluasi	7
1.6.6 Penyusunan Buku Tugas Akhir	7
1.7 Sistematika Penulisan Laporan Tugas Akhir	8

BAB II TINJAUAN PUSTAKA	11
2.1 Sistem Informasi Akademik Generik	11
2.2 Perangkat Lunak Modular	11
2.3 Model-View-Controller	12
2.4 Hierarchical Model-View-Controller (HMVC)	13
2.5 Repository Service Pattern	14
2.6 PHP (Hypertext Preprocessor)	14
2.7 Framework Phalcon	15
2.8 SQL Server	16
2.9 OAuth2	16
2.10 JSON Web Token (JWT)	16
2.11 Role Based Access Control (RBAC)	17
2.12 RESTful API	17
BAB III ANALISIS DAN PERANCANGAN PERANGKAT LUNAK	19
3.1 Analisis	19
3.1.1 Analisis Permasalahan	19
3.1.2 Deskripsi Umum Sistem	23
3.1.3 Kebutuhan Fungsional	23
3.1.4 Kasus Penggunaan	24
3.1.5 Kebutuhan Non Fungsional	34
3.1.6 Ketergantungan Data Antar Modul	34
3.2 Perancangan Perangkat Lunak	35
3.2.1 Desain Arsitektur	36

3.2.2	Mekanisme Login.....	41
3.2.3	RBAC	47
3.2.4	Restful API.....	48
3.2.5	Routing	52
3.2.6	Internationalization (i18n).....	53
3.2.7	Logging.....	55
3.2.8	Push Notification	56
3.2.9	Konvensi Kode	59
3.2.10	Perancangan Antar Muka	62
BAB IV IMPLEMENTASI		65
4.1	Lingkungan Implementasi	65
4.2	Implementasi Modularitas	66
4.3	Implementasi Routing	73
4.4	Implementasi Repository Service Pattern	78
4.5	Implementasi Modul Service Pada Kerangka Kerja.....	81
4.5.1	Implementasi OAuth2	81
4.5.2	Implementasi RBAC	90
4.5.3	Implementasi Internationaliation (i18n)	96
4.5.4	Implementasi Logging.....	102
4.5.5	Implementasi API.....	106
4.5.6	Implementasi Push Notification	109
4.5.7	Implementasi Modul Pengaturan.....	112
4.6	Implementasi Perancangan Antarmuka	114

4.6.1	Antarmuka Halaman Login SIAKAD	114
4.6.2	Antarmuka Halaman <i>Dashboard</i> Untuk Mengganti Bahasa	116
4.6.3	Antarmuka Halaman <i>Dashboard</i> Untuk Mengganti Hak Akses.....	116
4.6.4	Antarmuka Halaman Modul Pengaturan.....	117
4.7	Implementasi Pada Server.....	125
BAB V UJI COBA DAN EVALUASI		129
5.1	Lingkungan Uji Coba	129
5.2	Skenario Pengujian	129
5.2.1	Kasus Pengujian Oauth2	130
5.2.2	Kasus Pengujian RBAC	132
5.2.3	Kasus Pengujian Internationalization	136
5.2.4	Kasus Pengujian Logging.....	140
5.2.5	Kasus Pengujian API.....	142
5.2.6	Kasus Pengujian Push Notification	145
5.2.7	Kasus Pengujian Modul Pengaturan.....	146
BAB VI KESIMPULAN DAN SARAN.....		153
6.1	Kesimpulan	153
6.2	Saran.....	154
DAFTAR PUSTAKA		155
BIODATA PENULIS.....		157

DAFTAR GAMBAR

Gambar 1.1 Diagram Sistem Informasi Akademik Modular.....	2
Gambar 2.1 Diagram MVC + Repository Service Pattern	12
Gambar 2.2 Diagram HMVC	13
Gambar 2.3 Diagram <i>Framework</i> Terpopuler	15
Gambar 3.1 Proses tahapan membangun SIAKAD	19
Gambar 3.2 Kerangka kerja SIAKAD	23
Gambar 3.3 Diagram kasus penggunaan.....	26
Gambar 3.4 Diagram aktivitas mengelola modul pengaturan	28
Gambar 3.5 Diagram aktivitas memilih bahasa pengantar	30
Gambar 3.6 Diagram aktivitas mengganti hak akses pengguna..	33
Gambar 3.7 Ketergantungan data antar modul	35
Gambar 3.8 Desain arsitektur SIAKAD modular	37
Gambar 3.9 Struktur direktori pada folder modules.....	39
Gambar 3.10 Struktur direktori pada modul.....	40
Gambar 3.11 Arsitektur Repository Service Pattern pada SIAKAD	41
Gambar 3.12 Diagram Controller pada SIAKAD	41
Gambar 3.13 <i>Flowchart</i> mekanisme login.....	43
Gambar 3.14 <i>flowchart</i> authorization code	45
Gambar 3.15 Spesifikasi tabel menu pada myITS	47
Gambar 3.16 Diagram mekanisme RBAC	48
Gambar 3.17 Mekanisme API pada SIAKAD	52
Gambar 3.18 <i>Flowchart</i> i18n	54
Gambar 3.19 Mekanisme <i>logging</i>	55
Gambar 3.20 Spesifikasi tabel log pada myITS.....	56
Gambar 3.21 <i>flowchart web push notification</i> untuk <i>device</i> token	57
Gambar 3.22 Spesifikasi tabel <i>device</i> token pada myITS	58

Gambar 3.23 Mekanisme pengiriman pesan pada fitur <i>push notification</i>	58
Gambar 3.24 Penulisan kode PHP menggunakan PSR-1 dan PSR-2	60
Gambar 3.25 Konvensi kode JavaScript.....	62
Gambar 3.26 Perancangan antar muka untuk halaman login.....	62
Gambar 3.27 Perancangan antar muka untuk memilih bahasa..	63
Gambar 3.28 Perancangan antar muka untuk memilih opsi hak akses	63
Gambar 3.29 Perancangan antar muka 404 not found.....	64
Gambar 3.30 Perancangan antar muka untuk modul pengaturan	64
Gambar 4.1 struktur modul penilaian	68
Gambar 4.2 halaman baseUrl.....	75
Gambar 4.3 halaman baseUrl/frs/jadwal-kuliah	76
Gambar 4.4 folder models.....	79
Gambar 4.5 antar muka halaman menambahkan data user	91
Gambar 4.6 antar muka halaman menambahkan menu pada myITS	92
Gambar 4.7 dashboard pengguna dosen	93
Gambar 4.8 dashboard pengguna mahasiswa	93
Gambar 4.9 folder language pada SIAKAD	98
Gambar 4.10 sidebar (a) bahasa Inggris; (b) bahasa indonesia	101
Gambar 4.11 folder library logger	102
Gambar 4.12 Pesan <i>pop up</i> pada fitur <i>push notification</i>	112
Gambar 4.13 file setting.json	113
Gambar 4.14 modul tanpa file setting.json	113
Gambar 4.15 struktur direktori modul pengaturan	114
Gambar 4.16 halaman login SIAKAD	115
Gambar 4.17 halaman login myITS.....	115

Gambar 4.18 antarmuka halaman mengganti bahasa pengantar	116
Gambar 4.19 antarmuka halaman mengganti hak akses pengguna	117
Gambar 4.20 antarmuka halaman modul pengaturan.....	118
Gambar 4.21 antarmuka halaman modul pengaturan umum ..	118
Gambar 4.22 modal untuk mengubah data modul pengaturan umum	119
Gambar 4.23 antarmuka halaman modul pengaturan FRS	120
Gambar 4.24 modal untuk mengubah data modul pengaturan FRS	120
Gambar 4.25 antarmuka halaman modul pengaturan IPD	121
Gambar 4.26 modal untuk mengubah data modul pengaturan IPD	121
Gambar 4.27 antarmuka halaman modul pengaturan penilaian	122
Gambar 4.28 modal untuk mengubah data modul pengaturan penilaian	122
Gambar 4.29 antarmuka halaman modul pengaturan SAR	123
Gambar 4.30 modal untuk mengubah data modul pengaturan SAR.....	123
Gambar 4.31 antarmuka halaman modul pengaturan yudisium	124
Gambar 4.32 modal untuk mengubah data modul pengaturan yudisium	124
Gambar 4.33 folder SIAKAD di server.....	127
Gambar 4.34 folder SIAKAD di <i>symlink</i> ke <i>/var/www/siakad...</i>	127
Gambar 5.1 hasil pengujian skenario 1 pada kasus pengujian RBAC	135
Gambar 5.2 hasil pengujian skenario 2 pada kasus pengujian RBAC	135

Gambar 5.3 hasil pengujian skenario 3 pada kasus pengujian RBAC	136
Gambar 5.4 hasil pengujian skenario 1 pada kasus pengujian <i>internationalization</i>	139
Gambar 5.5 hasil pengujian skenario 2 pada kasus pengujian <i>internationalization</i>	139
Gambar 5.6 hasil pengujian skenario 2 pada kasus pengujian <i>internationalization</i>	140
Gambar 5.7 Hasil pengujian skenario 1 pada API.....	144
Gambar 5.8 Hasil pengujian skenario 2 pada API.....	144
Gambar 5.9 Hasil pengujian skenario 1 pada fitur <i>push notification</i>	146

DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional	24
Tabel 3.2 Kelompok pengguna SIAKAD	25
Tabel 3.3 Spesifikasi kasus penggunaan mengelola modul pengaturan	26
Tabel 3.4 Spesifikasi kasus penggunaan memilih bahasa pengantar	28
Tabel 3.5 Spesifikasi kasus penggunaan mengganti hak akses pengguna	31
Tabel 3.6 Kebutuhan non fungsional.....	34
Tabel 3.7 Contoh JWT yang diterapkan.....	49
Tabel 3.8 JWT yang telah di decode	51
Tabel 3.9 Parameter format routing	53
Tabel 4.1 Spesifikasi lingkungan pengembangan	65
Tabel 4.2 Spesifikasi lingkungan produksi	66
Tabel 4.3 Kriteria ClassName	67
Tabel 4.4 Parameter Url pada endpoint /authorize	83
Tabel 4.5 Deskripsi parameter pada Url endpoint /authorize	83
Tabel 5.1 Lingkungan Pengujian Klien	129
Tabel 5.2 Kasus Pengujian Oauth2	130
Tabel 5.3 Kasus Pengujian RBAC.....	132
Tabel 5.4 Kasus pengujian <i>internationalization</i>	137
Tabel 5.5 Kasus pengujian logging.....	140
Tabel 5.6 Kasus pengujian API.....	142
Tabel 5.7 Kasus pengujian pada fitur <i>push notification</i>	145
Tabel 5.8 Kasus pengujian modul pengaturan	147

[halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode sumber 4.1 mendaftarkan modul pada loader.php	67
Kode sumber 4.2 mendaftarkan module.php pada modules.php	67
Kode sumber 4.3 mendefinisikan controllers dan views module	69
Kode sumber 4.4 namespace pada controller penilaian	70
Kode sumber 4.5 namespace pada controller sar	70
Kode sumber 4.6 file ControllerBase	72
Kode sumber 4.7 fungsi mengecek hak akses	73
Kode sumber 4.8 implementasi routing	74
Kode sumber 4.9 memanggil kelas service pada controller	77
Kode sumber 4.10 memanggil kelas repository pada service	77
Kode sumber 4.11 raw query pada controller	78
Kode sumber 4.12 file service pada kelas Aktivitas	80
Kode sumber 4.13 <i>repository</i> pada kelas Skem	80
Kode sumber 4.14 fungsi meminta akses token	86
Kode sumber 4.15 parameter Url pada config.php	87
Kode sumber 4.16 service untuk mengirimkan parameter ke endpoint /token	87
Kode sumber 4.17 fungsi meminta informasi pengguna	88
Kode sumber 4.18 service meminta informasi pengguna pada endpoint /userinfo	88
Kode sumber 4.19 menyimpan informasi pengguna ke session ..	89
Kode sumber 4.20 fungsi mengarahkan pengguna ke dashboard	90
Kode sumber 4.21 Implementasi untuk mengarahkan pengguna ke dashboard sesuai hak akses	92
Kode sumber 4.22 informasi pengguna pada API myITS	95
Kode sumber 4.23 informasi menu pengguna pada myITS	96

Kode sumber 4.24 fungsi menyimpan informasi bahasa pengantar	97
Kode sumber 4.25 file terjemahan bahasa Indonesia.....	99
Kode sumber 4.26 file terjemahan bahasa inggirs	100
Kode sumber 4.27 mencetak variabel bahasa ke tampilan.....	100
Kode sumber 4.28 mencatat log aktivitas	103
Kode sumber 4.29 fungsi untuk insert log.....	104
Kode sumber 4.30 data log.....	105
Kode sumber 4.31 fungsi untuk mengirimkan log	105
Kode sumber 4.32 versi API.....	106
Kode sumber 4.33 mendaftarkan modul API	106
Kode sumber 4.34 routing modul API	107
Kode sumber 4.35 contoh endpoint API	108
Kode sumber 4.36 fungsi validasi id token.....	108
Kode sumber 4.37 fungsi menangkap token pada header.....	109
Kode sumber 4.38 Konfigurasi firebase cloud messaging	110
Kode sumber 4.39 Inisialisasi penggunaan firebase.....	110
Kode sumber 4.40 Fungsi untuk meminta izin fitur <i>push notification</i>	110
Kode sumber 4.41 Fungsi untuk mendapatkan token	111
Kode sumber 4.42 status pengiriman <i>push notification</i>	111
Kode sumber 4.43 konfigurasi file SIAKAD di nginx	127

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem Informasi Akademik (SIKAD) pada perguruan tinggi dirancang untuk memudahkan dalam pengelolaan informasi seperti pengelolaan nilai mahasiswa, pengambilan matakuliah, penjadwalan matakuliah, menampilkan indeks prestasi semester, menampilkan prestasi kumulatif dan proses-proses yang berkaitan dengan akademik lainnya yang terdapat di perguruan tinggi.

Pada saat ini Direktorat Pengembangan Teknologi dan Sistem Informasi (DPTSI) ITS sedang melakukan perubahan Sistem Informasi Akademik (SIKAD) secara besar-besaran. Hal tersebut dilakukan karena sistem yang lama sudah tidak sesuai dengan keadaan pada saat ini. Banyak perubahan proses bisnis yang terjadi di kampus ITS sehingga mengakibatkan fungsionalitas yang terdapat pada SIKAD harus diubah dan diganti dengan yang baru.

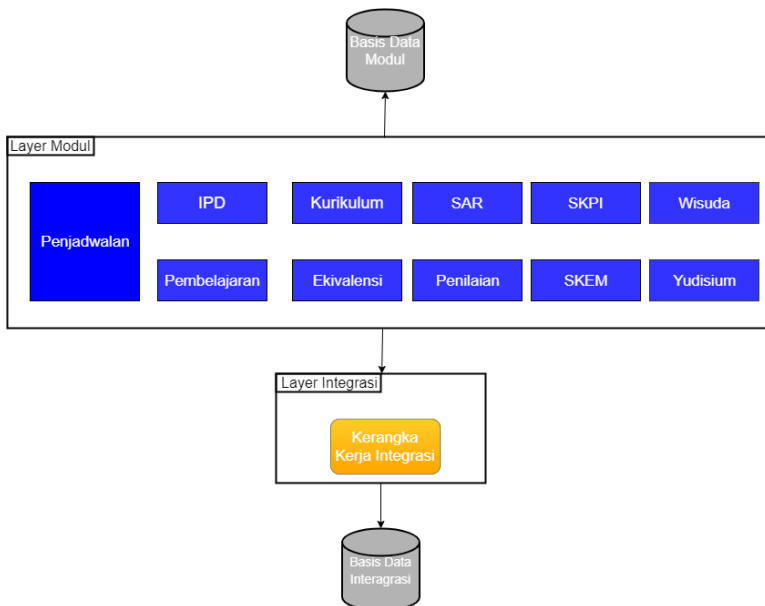
Pada saat ini, SIKAD yang dimiliki oleh kampus ITS untuk dapat memenuhi perubahan-perubahan proses bisnis dan kebutuhan-kebutuhan yang diperlukan sangat sulit untuk dikembangkan, hal tersebut dikarenakan SIKAD yang dimiliki ITS tidak dirancang dan dibangun dengan sistem yang modular, sehingga menyulitkan bagi pengembang untuk melakukan pengembangan terhadap sistem dan memakan banyak waktu untuk mempelajari sistem dan melakukan perubahan.

Setiap Perguruan tinggi, termasuk kampus ITS, memiliki banyak kesamaan dalam fungsionalitas yang terdapat pada SIKAD masing-masing perguruan tinggi. Hal tersebut dapat dijadikan landasan untuk adanya perancangan dan pembangunan SIKAD generik yang bersifat modular.

SIKAD generik merupakan sistem yang memiliki fungsionalitas umum yang dapat digunakan pada perguruan tinggi lainnya [1]. Sehingga dengan adanya SIKAD generik ini dapat meminimalisasi tingkat permintaan dan perulangan terhadap perancangan dan pembangunan SIKAD. Akan tetapi tidak

menutup kemungkinan terdapat perbedaan fungsionalitas yang dibutuhkan antar perguruan tinggi sehingga untuk mengatasi perbedaan fungsionalitas tersebut dibutuhkan SIAKAD yang bersifat modular.

SIAKAD bersifat modular yaitu sistem yang pada perancangan dan pembangunannya dipecah atas beberapa modul sehingga perguruan tinggi dapat memilih modul yang diperlukan yang tidak sesuai dengan fungsionalitas yang berlaku. Hal ini juga mengakibatkan sistem dengan mudah untuk diperbarui apabila ada perubahan proses bisnis dan penambahan modul yang diinginkan. Gambar 1.1 menunjukkan dari SIAKAD generik bersifat modular yang akan dibangun.



Gambar 1.1 Diagram Sistem Informasi Akademik Modular

Pada Gambar 1.1 terdapat 11 modul pada SIAKAD yang akan dibangun diantaranya kurikulum, ekivalensi, penjadwalan, IPD, pembelajaran (FRS), dst. Untuk Tugas Akhir ini, pengerjaan hanya dilakukan pada kerangka kerja integrasi sedangkan modul lainnya akan dikerjakan pada Tugas Akhir lainnya.

Pada penelitian Tugas Akhir sebelumnya, pembangunan dan perancangan SIAKAD generik bersifat modular sudah terealisasi akan tetapi terdapat banyak kekurangan dalam implementasinya yaitu terdapat banyak *hardcode* dalam menentukan *path* (lokasi) untuk modul-modul yang akan diimplementasikan ke dalam sistem, sehingga membuat sistem menjadi tidak *dinamis*. Hal tersebut dikarenakan untuk menentukan *path* dari modul harus sesuai dengan *hardcode*, hal ini akan menjadi rumit apabila sistem akan dipasang di lingkungan instalasi yang berbeda. Kekurangan selanjutnya adalah tidak adanya standar keamanan yang digunakan sehingga sistem tidak memiliki keamanan yang terjamin dan akan menyulitkan bagi pengembangan selanjutnya, dan tidak menerapkan RBAC (*Role Based Access Control*) ke dalam sistem.

Oleh karena itu, pada Tugas Akhir ini akan membangun ulang SIAKAD dengan perancangan dan kebutuhan yang baru. Adapun yang akan dikembangkan dalam Tugas Akhir ini adalah adanya pengembangan dari sisi keamanan yaitu menggunakan standar keamanan untuk otorisasi dengan protokol OAuth2, API, menerapkan JWT (*JSON Web Token*) untuk mengakses API, mengimplementasikan RBAC ke sistem, adanya *push notification* terhadap sistem, *Logging*, *internationalization*, dan *Routing*.

1.2 Rumusan Masalah

Rumusan masalah pada Tugas Akhir ini adalah sebagai berikut.

1. Bagaimana mengadopsi dan memperbaharui SIAKAD yang telah dikembangkan sebelumnya?
2. Bagaimana membuat kerangka kerja SIAKAD modular?
3. Bagaimana menerapkan RBAC pada SIAKAD?

4. Bagaimana menerapkan metode *JSON Web Token* (JWT) dan protokol OAuth2 untuk melakukan otorisasi pada SIAKAD?
5. Bagaimana penerapan *routing* pada SIAKAD?
6. Bagaimana menerapkan *logging*, *API*, *Internationalization*, dan *push notification* pada SIAKAD?

1.3 Batasan Masalah

Batasan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut.

1. Kerangka kerja dibuat menggunakan bahasa pemrograman PHP dengan *Framework* Phalcon.
2. Manajemen basis data yang digunakan yaitu SQL Server 2017.
3. Pengerjaan kerangka kerja meliputi pengerjaan pembuatan rancang bangun utama dari sistem, pembuatan aturan-aturan modularitas, basis data, dan keamanan.
4. Sistem yang dikerjakan berbasis *web*.
5. Pengerjaan kerangka kerja tidak termasuk pengerjaan modul-modul yang terdapat pada SIAKAD generik bersifat modularitas dikarenakan akan dikerjakan pada Tugas Akhir lainnya.

1.4 Tujuan

Tujuan pembuatan Tugas Akhir ini adalah mengembangkan kerangka kerja pada sistem informasi akademik generik bersifat modular yang telah ada dan disesuaikan dengan kebutuhan yang baru, adapun pengembangan yang akan dilakukan adalah seperti berikut.

1. Menentukan RBAC (*Role Based Access Control*) pada sistem
2. Menerapkan pola arsitektur *Model View Controller* dan *Repository Service Pattern*
3. Menerapkan JWT pada API
4. Menerapkan protokol OAuth2 pada sistem untuk otorisasi
5. Menentukan penerapan *routing* pada sistem

6. Menerapkan *logging*, *API*, *internationalization*, dan *push notification* pada sistem

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah terciptanya kerangka kerja yang modular dan generik dari SIAKAD sehingga menjadi suatu sistem yang mudah untuk dikembangkan di masa yang akan datang dan dapat disesuaikan dengan fungsionalitas pada masing-masing perguruan tinggi sehingga sistem ini dapat dipakai oleh berbagai perguruan tinggi di Indonesia. Sistem yang dibangun memiliki standar keamanan yang baik yaitu menerapkan OAuth2 untuk memberi sebuah token apabila pengguna telah berhasil *login* ke sistem, dan sistem juga menyediakan mekanisme untuk mencatat log aplikasi, *internationalization*, *push notification*, dan *API*.

1.6 Metodologi

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal Tugas Akhir ini berisi tentang perencanaan “Rancang bangun kerangka kerja sistem informasi akademik modular berbasis web dengan pola arsitektur *hierarchical model-view-controller* dan *repository service pattern*”.

Proposal Tugas Akhir ini terdiri dari deskripsi pendahuluan yang menjabarkan latar belakang dari sistem yang akan dibangun, rumusan masalah yang mendasari dibangunnya sistem ini, batasan masalah dalam pembangunan aplikasi ini, serta tujuan dan manfaat yang diharapkan dapat dicapai dengan dibangunnya aplikasi ini. Selain itu, pada proposal Tugas Akhir ini terdapat tinjauan pustaka yang menjelaskan teori-teori yang menjadi dasar pembuatan Tugas Akhir, ringkasan isi Tugas Akhir yang menggambarkan secara umum aplikasi yang akan dibangun dan *framework* yang digunakan, serta bagian metodologi dari penyusunan Tugas Akhir ini.

1.6.2 Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi dan pembelajaran yang akan digunakan pada Tugas Akhir ini. Studi literatur meliputi diskusi dan pemahaman mengenai topik Tugas Akhir ini, diantaranya mengenai:

1. Modularitas perangkat lunak
2. Pola Arsitektur *Hierarchical Model-View-Controller*
3. *Repository service pattern*
4. SQL Server 2017 sebagai basis data

1.6.3 Analisis dan Desain Perangkat Lunak

Adapun pembagian tahap analisa kebutuhan dan perancangan dari kerangka kerja sistem informasi akademik sebagai berikut:

1. Mempelajari kebutuhan kerangka kerja secara umum. Pada tahap ini yaitu mempelajari kebutuhan aplikasi kerangka kerja web secara umum melalui aplikasi serupa yang telah dikembangkan.
2. Mempelajari kebutuhan umum sistem informasi akademik secara garis besar. Tahap ini dilakukan agar dapat membuat rancang bangun kerangka kerja yang sesuai kebutuhan pada tahap selanjutnya.
3. Merancang kerangka kerja sistem informasi akademik. Pada tahap ini dilakukan perancangan struktur kelas dalam bentuk *class diagram* dan basis data dalam bentuk *physical data model* dan *conceptual data model*.

1.6.4 Implementasi Perangkat Lunak

Kerangka kerja sistem informasi akademik generik bersifat modularitas ini akan diimplementasikan dengan menggunakan:

1. Bahasa pemrograman PHP
2. Framework Phalcon
3. SQL Server 2017

1.6.5 Pengujian dan Evaluasi

Pengujian dilakukan kepada beberapa mahasiswa ITS dan dosen untuk mengetahui tingkat keberhasilan kerangka kerja dari sistem informasi akademik generik bersifat modular yang dibangun sudah berjalan dengan baik dan tidak ada *bug* yang terjadi. Metode yang digunakan yaitu metode pengujian *black box* terhadap masukan dan keluaran yang dihasilkan berdasarkan skenario yang telah ditentukan. Tim testing melakukan pengujian fungsionalitas saat fase testing pada *Software Development Life Cycle* (SDLC). Metode pengujian dapat diterapkan pada setiap dan seluruh tingkatan dari pengujian perangkat lunak seperti unit, integrasi sistem, dan pengujian kelayakan.

1.6.6 Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain.

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistem Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari sistem yang dikerjakan pada Tugas Akhir. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir ini terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini membahas mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu, rumusan masalah, Batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi tentang analisis dan perancangan yang dijelaskan dalam bentuk *flowchart*, diagram, dan gambar desain antar muka yang terdiri dari analisis permasalahan, deskripsi umum sistem, kebutuhan fungsional, kasus penggunaan, kebutuhan non fungsional, dan ketergantungan data antar modul.

Bab IV Implementasi

Bab ini membahas implementasi dari rancangan dan desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa langkah dalam mengimplementasikan sistem, *code* yang mengimplementasikan dan gambar-gambar hasil implementasi yang telah dilakukan. Implementasi yang dijelaskan meliputi lingkungan implementasi, implementasi modularitas, implementasi *routing*, implementasi *repository service pattern*, modul service layer, implementasi perancangan antarmuka, dan implementasi pad server.

Bab V Uji Coba dan Evaluasi

Bab ini akan menjelaskan bagaimana pengujian yang dilakukan terhadap sistem, pengujian pada setiap fitur-fitur yang telah diimplementasiikan seperti pengujian pada OAuth2, RBAC, *Internationalization*, *logging*, *API*, *push notification*, dan pengujian modul pengaturan serta memaparkan hasil-hasil pengujian yang telah dilakukan.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari sistem yang telah dibangun dan saran untuk pengembangan perangkat lunak ke depannya.

[halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi Akademik Generik

Sistem informasi akademik generik merupakan sistem dengan proses bisnis yang bersifat umum dan dapat digunakan untuk variasi proses bisnis akademik di perguruan tinggi lainnya [1]. Sistem informasi akademik ini terdiri dari beberapa modul yaitu.

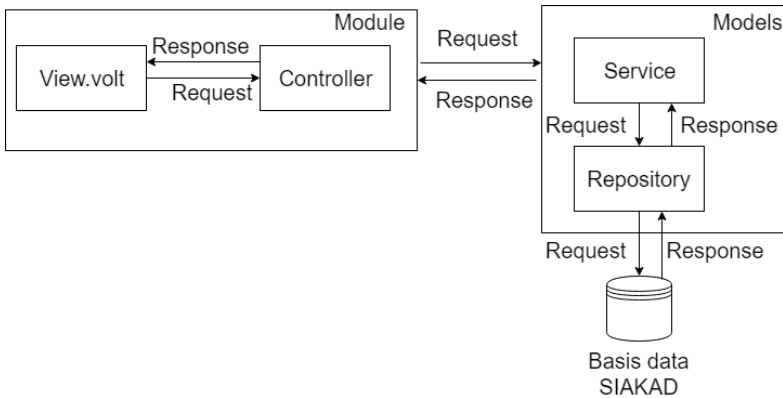
1. Modul pembelajaran dan FRS
2. Modul SAR dan penilaian
3. Modul wisuda dan yudisium
4. Modul SKEM dan SKPI
5. Modul kurikulum dan ekivalensi
6. Modul Penjadwalan

Masing-masing modul tersebut dikembangkan dalam setiap strukur folder yang terpisah. Modul-modul tersebut dikembangkan dengan pola *Model-View-Controller* dan menggunakan *Repository-Service-Pattern* dengan kerangka kerja Phalcon. Sebagai contoh arsitektur modul, Gambar 2.1 menunjukkan arsitektur yang digunakan pada setiap modul.

2.2 Perangkat Lunak Modular

Perangkat lunak modular adalah perangkat lunak yang terbagi menjadi beberapa modul dimana modul-modul tersebut fokus pada satu fungsionalitas. Dengan perancangan modular, suatu perangkat lunak memiliki kelebihan seperti.

1. Perangkat lunak dapat dipelajari per modul sehingga lebih mudah dipahami.
2. Modul dapat lebih mudah dikelola maupun diperbaiki jika ada penambahan ataupun kekurangan pada modul tersebut.



Gambar 2.1 Diagram MVC + Repository Service Pattern

2.3 Model-View-Controller

MVC merupakan salah satu dari *design pattern* yang terdiri dari *Model*, *View*, dan *Controller*. *Design pattern* tersebut digunakan untuk memisahkan antara *view* (tampilan) dengan *model* (data) dan *Controller* (yang menghubungkan antara *controller* dan *model*) [2]. Detail penjelasan dari MVC adalah sebagai berikut.

1. *Model*

Model berisi fungsi-fungsi yang berhubungan langsung dengan basis data untuk mengelola data seperti menambahkan data, menghapus data, mengubah data, dan menampilkan data.

2. *View*

View adalah bagian yang menangani tampilan (*user interface*) ke pengguna dan merepresentasikan data yang terdapat pada model.

3. *Controller*

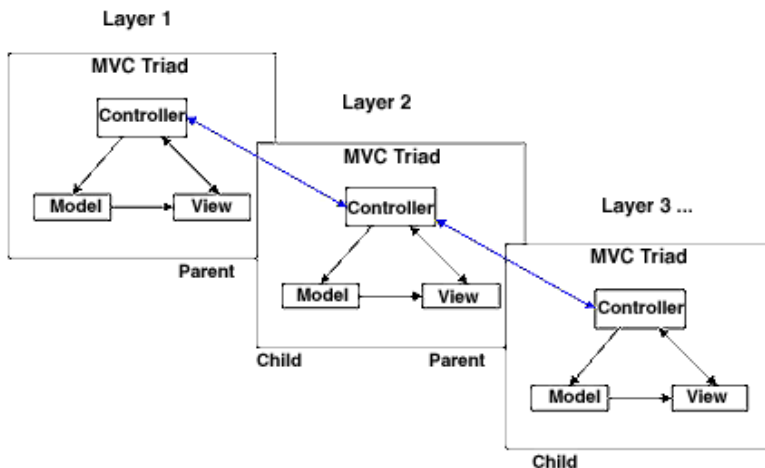
Controller merupakan bagian yang menghubungkan antara *model* dan *view*. *Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

2.4 Hierarchical Model-View-Controller (HMVC)

HMVC merupakan pengembangan dari design pattern MVC yang sering digunakan sebagian besar untuk membangun aplikasi web saat ini [3]. HMVC adalah versi MVC yang diimplementasikan secara hirarkis. Hirarkis di sini maksudnya adalah hirarki folder (struktur folder) yang mana menjadi modul [4]. Adapun kelebihan menggunakan HMVC adalah sebagai berikut:

1. *Modularization*
Mengurangi ketergantungan (dependensi) antara bagian-bagian yang berbeda dari aplikasi.
2. *Organization*
Setiap modul memiliki MVC tersendiri sehingga membuat beban kerja aplikasi menjadi lebih ringan
3. *Reusability*
Mudah untuk diperbaiki (maintenance) setiap baris kode
4. *Extendibility*
Membuat aplikasi mudah untuk dikembangkan

Gambar 2.2 menunjukkan diagram HMVC.



Gambar 2.2 Diagram HMVC

2.5 Repository Service Pattern

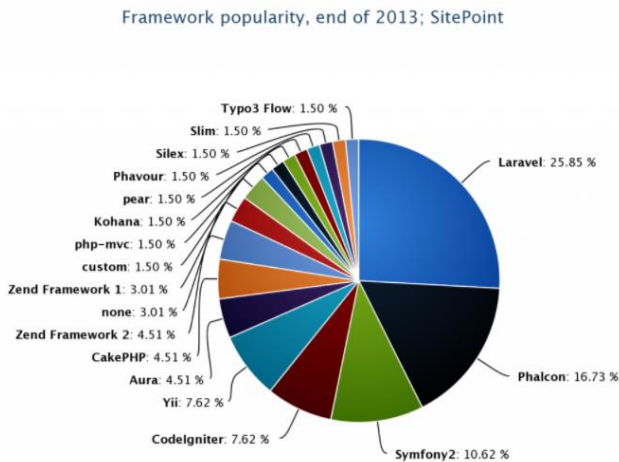
Penggunaan *pattern* ini adalah untuk menerapkan konsep pemisahan kode program berdasarkan fungsinya. Semua kode untuk akses database harus dipisahkan dengan kode untuk pengaturan *user interface*. Hal ini memungkinkan kode akses database yang dibuat untuk aplikasi desktop, dengan mudah digunakan untuk aplikasi web. *Repository Pattern* berisi semua kode untuk mengakses database. Semua kode yang spesifik terhadap implementasi akses database berhenti di sini, lapisan lebih atas tidak boleh tahu bagaimana akses database diterapkan, apakah menggunakan ADO.NET murni atau tool ORM/Micro ORM seperti Dapper.NET, Entity Framework atau Hibernate. Lapisan lainnya hanya perlu tahu fungsionalitas dari suatu method di dalam class Repository, tidak perlu tahu bagaimana method tersebut diimplementasikan [5].

2.6 PHP (Hypertext Preprocessor)

PHP atau singkatan dari *Hypertext Preprocessor* merupakan salah satu bahasa pemrograman yang banyak digunakan untuk pengembangan sebuah web dan dapat disematkan ke dalam HTML [6]. PHP fokus pada scripting dari sisi server (server-side scripting), sehingga developer dapat melakukan apa saja yang dapat dilakukan oleh program CGI lainnya, seperti mengumpulkan data formulir, menghasilkan konten halaman dinamis, atau mengirim dan menerima *cookies*. PHP dapat digunakan pada semua sistem operasi seperti linux, varian Unix (termasuk HP-UX, solaris dan OpenBSD), Windows, Mac OS, RISC S, dan sistem operasi lainnya. dengan menggunakan PHP, developer memiliki kebebasan untuk memilih sistem operasi. Selain itu, PHP juga dapat mendukung pemrograman yang prosedural ataupun pemrograman berorientasi object (OOP) atau bahkan campuran antara prosedural dan OOP [7].

2.7 Framework Phalcon

Phalcon merupakan salah satu framework untuk bahasa pemrograman PHP. Framework ini ditulis dalam bahasa C, C++, dan PHP. Phalcon juga menggunakan pola MVC seperti halnya framework lainnya seperti Laravel, dan CodeIgniter. Phalcon mendukung ORM, Pagination, Cache, Form builder dan Template engine yang memiliki nama volt dan ekstensi (.volt). Diakhir tahun 2013 survei yang diadakan oleh sitepoint.com Framework phalcon menduduki peringkat kedua sebagai framework terpopuler setelah Laravel yang menduduki peringkat pertama [8]. Gambar 2.3 Berikut merupakan diagram survei yang diadakan oleh sitepoint.com.



Gambar 2.3 Diagram *Framework* Terpopuler

2.8 SQL Server

SQL Server merupakan sistem manajemen basis data relasional (RDBMS) produk Microsoft. bahasa kueri yang digunakan adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase [9]. Dalam Tugas Akhir ini SQL Server yang digunakan merupakan SQL Server versi 2017 Express Edition.

2.9 OAuth2

OAuth2 (Open Authorization) yang banyak digunakan oleh developer sebagai proses authorization sebuah aplikasi. Dengan menggunakan protokol OAuth2 maka memungkinkan aplikasi ketiga untuk dapat mengakses data dari aplikasi yang telah kita bangun. ada beberapa grant type pada OAuth2 yaitu: grant type authorization code, grant type owner password, grant type client credentials, dan grant type implicit client [10].

2.10 JSON Web Token (JWT)

JWT merupakan standar terbuka (RFC 7519) yang mendefinisikan cara yang kompak dan bisa dilihat pada isinya untuk mentransmisikan informasi secara aman antar berbagai pihak sebagai obyek JSON. Adapun struktur dari JWT terdiri atas 3 bagian, dipisahkan oleh tanda titik (.): xxx.yyy.zzz [10]. Berikut penjelasan dari masing-masing bagian struktur pada JWT.

1. *Header* (xxx): {"typ": "JWT", "alg": "HS256"}
2. *Payload* (yyy) merupakan JWT claims, bisa berupa 3 claims: registered, public, dan private claims. Registered terdiri atas.
 - a. iss: issuer dari token
 - b. sub: subject dari token
 - c. aud: audience dari token
 - d. exp: waktu expired dari token, harus lebih dari tanggal sekarang
 - e. nbf: waktu sebelum JWT diterima untuk proses
 - f. iat: waktu JWT dikeluarkan

- g. jti: identifier unik untuk JWT, ada dua jenis yaitu public dan private
- 3. Signature (zzz) berisi header + payload + secret pas, dienkripsi menggunakan algoritma yang terdapat pada header (nilai dari alg).

2.11 Role Based Access Control (RBAC)

RBAC merupakan pembagian akses terhadap aplikasi yang disesuaikan dengan tingkatan pengguna. Dengan metode RBAC dapat memberikan keamanan karena dapat membatasi akses pengguna terhadap sistem, hal ini mencegah pengguna untuk mengakses informasi yang tidak relevan bagi mereka. Dalam konteks ini, pembatasan akses merupakan pembatasan untuk melakukan tugas tertentu, seperti melihat, membuat, dan memodifikasi file [11].

2.12 RESTful API

REST adalah singkatan dari REpresentational State Transfer yang merupakan arsitektur standar dari web dan menggunakan protokol HTTP untuk melakukan komunikasi data. Metode yang paling sering digunakan dalam arsitektur REST adalah sebagai berikut [12].

1. GET – untuk mengambil atau membaca data dari resources
2. PUT – untuk membuat atau menambahkan data ke resources
3. DELETE – untuk menghapus data pada resources
4. POST – untuk menambahkan dan mengubah data pada resources

Sedangkan API adalah singkatan dari Application Programming Interface yang merupakan suatu penghubung untuk berinteraksi antara suatu aplikasi dengan aplikasi lainnya. Output dari API

adalah JSON dan XML, pada Tugas Akhir ini output yang dihasilkan dalam bentuk JSON [13].

BAB III

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

Pada bab ini membahas analisis dan perancangan perangkat lunak dari kerangka kerja sistem informasi akademik (SIKAD). Hasil dari proses ini berupa diagram yang akan digunakan sebagai acuan untuk proses implementasi perangkat lunak, diagram hasil dari proses perancangan digunakan sebagai dokumentasi dari implementasi perangkat lunak. Diagram yang dihasilkan pada proses ini disajikan dalam bentuk *Unified Modelling Language* (UML). Gambar 3.1 menunjukkan tahapan-tahapan yang dilakukan untuk membangun SIKAD dari proses analisis sampai tahapan konstruksi perangkat lunak.



Gambar 3.1 Proses tahapan membangun SIKAD

3.1 Analisis

Analisis meliputi analisis permasalahan, pendeskripsian perangkat lunak secara umum, menjelaskan ketergantungan antara setiap modul, dan penjelasan mekanisme setiap fitur-fitur yang dikerjakan dengan menggunakan diagram.

3.1.1 Analisis Permasalahan

Suatu SIKAD pada Perguruan Tinggi umumnya digunakan untuk mengolah semua proses bisnis yang terkait dengan akademik seperti dapat menampilkan IPK, mengambil matakuliah, melihat transkrip, melihat jadwal kuliah, dll. Dengan

adanya SIAKAD proses-proses yang berkaitan dengan akademik pada perguruan tinggi menjadi lebih mudah, sistematis, dan terkomputerisasi.

Setiap perguruan tinggi memiliki kebutuhan berbeda-beda pada SIAKAD yang dimilikinya, hal tersebut tergantung bagaimana proses bisnis terkait akademik yang terdapat pada perguruan tinggi. Proses bisnis terkait akademik pada perguruan tinggi tentunya akan mengalami perubahan-perubahan kebutuhan seiring berjalannya waktu.

Pada SIAKAD yang dimiliki oleh Perguruan Tinggi ITS, perubahan-perubahan terhadap SIAKAD dilakukan seiring bertambahnya kebutuhan ataupun perubahan-perubahan proses bisnis yang terjadi di perguruan tinggi ITS, sehingga mengakibatkan sistem harus dikembangkan lagi untuk menyesuaikan dengan kebutuhan-kebutuhan dan proses bisnis yang baru.

Pada saat ini, SIAKAD pada perguruan tinggi ITS sudah mengalami perbaikan dan pembaharuan untuk dapat memenuhi kebutuhan dan menyesuaikan dengan proses bisnis terkait akademik yang baru. Akan tetapi masalah yang muncul adalah SIAKAD perguruan tinggi ITS tidak menerapkan perancangan sistem yang modular sehingga pengembangan selanjutnya akan sulit untuk dilakukan, dipelajari, dan sistem akan lebih tidak teratur dalam pengembangannya.

Permasalahan lain yang juga muncul pada perguruan tinggi ITS memiliki banyak sekali aplikasi-aplikasi selain SIAKAD. Aplikasi tersebut merupakan aplikasi yang digunakan untuk mempermudah kebutuhan-kebutuhan yang ada pada perguruan tinggi ITS. Permasalahannya adalah semua aplikasi tersebut tidak saling terintegrasi dengan baik. Kekurangan ini mengakibatkan banyak sekali data yang terduplikasi dan ketidaksinambungan data antara satu aplikasi dengan aplikasi lainnya.

Pada Tugas Akhir “Rancang bangun kerangka kerja sistem informasi akademik modular berbasis web dengan pola arsitektur Hierarchical Model-View-Controller“ [1] telah menerapkan

perancangan SIAKAD yang modular dengan menggunakan framework Spring MVC dan lingkungan implementasi yang mendukung OSGi. Implementasi pada Tugas Akhir sebelumnya [1] memiliki beberapa kekurangan diantaranya adalah tingkat dependensi yang tinggi, konflik antar dependensi, serta konfigurasi yang sangat rumit untuk dapat mengembangkan kerangka kerja.

Untuk dapat menangani permasalahan-permasalahan tersebut, maka akan dilakukan perancangan SIAKAD baru yang mengacu kepada kebutuhan-kebutuhan pada SIAKAD ITS yang lama sehingga dihasilkan suatu kesimpulan yaitu membangun sistem akademik modular dengan menggunakan framework Phalcon yang mendukung bahasa pemrograman PHP sehingga sistem akan menjadi lebih mudah untuk dipelajari, dan dikembangkan.

Fungsionalitas dan proses yang terdapat pada SIAKAD lama akan digunakan pada SIAKAD baru dengan perancangan modular sehingga rancang bangun aplikasi dapat dipecah menjadi beberapa modul, dengan perancangan yang modular aplikasi dengan mudah dapat dipelajari per modul, dan aplikasi menjadi lebih teratur dalam pengembangannya. Penambahan, penghapusan, dan perubahan pada modul dapat dilakukan dengan mudah.

Adapun modul-modul untuk memenuhi fungsionalitas perguruan tinggi ITS yang akan dibangun pada SIAKAD baru adalah modul penjadwalan, penilaian, SAR, FRS, IPD, SKEM, SKPI, kurikulum, ekivalensi, yudisium, dan wisuda. Fitur lainnya sebagai fitur pendukung pada SIAKAD adalah adanya penerapan *Oauth2*, *logging*, *API*, *push notification*, dan *Internationalization*, yang pada tugas akhir sebelumnya fitur-fitur tersebut belum diimplementasikan.

SIAKAD baru yang dibangun akan diterapkan sebagai klien dari aplikasi myITS. myITS merupakan aplikasi yang menerapkan *Single Sign On*, sehingga nantinya semua aplikasi yang terdapat pada ITS akan terhubung ke aplikasi myITS untuk proses otorisasi. Dengan adanya myITS pengguna dapat masuk ke semua aplikasi yang ada di ITS dengan sekali *login* dengan catatan pengguna

terdaftar ke aplikasi dan memiliki hak akses ke aplikasi tersebut. untuk dapat terhubung dengan aplikasi myITS maka aplikasi yang dibangun, termasuk SIAKAD, akan menggunakan protokol OpenID untuk proses autentikasi dan Oauth2 sebagai proses untuk otorisasi.

SIAKAD yang baru juga menyediakan *logging*. *Logging* diterapkan pada aplikasi untuk mencatat aktivitas-aktivitas yang dilakukan pengguna terhadap sistem sehingga untuk dapat melakukan audit pada sistem menjadi lebih mudah.

Dikarenakan pada saat ini terdapat banyak mahasiswa asing yang berasal dari luar Indonesia maka aplikasi yang dibangun harus dapat menyediakan mekanisme *internationalization*. Dengan adanya *internationalization* maka SIAKAD akan dibangun dengan menyediakan dua pilihan bahasa pengantar yaitu bahasa indonesia dan bahasa Inggris agar pengguna yang menggunakan aplikasi SIAKAD menjadi lebih mudah dan paham dalam menggunakannya. Penyediaan bahasa pengantar pada SIAKAD nantinya dapat dikembangkan lagi menjadi lebih banyak tergantung dengan kebutuhan pada perguruan tinggi kedepannya.

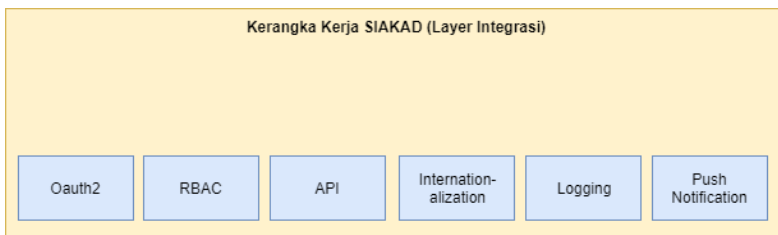
Banyak aplikasi yang terdapat di ITS, sehingga tidak menutup kemungkinan antara satu aplikasi dengan aplikasi lainnya perlu melakukan pertukaran data, untuk dapat menangani hal tersebut maka SIAKAD yang baru menyediakan API yang dapat diakses dengan menggunakan JWT token sebagai keamanan, apabila token tersebut valid maka pertukaran data dapat dilakukan dan apabila token tersebut tidak valid maka pertukaran data tidak dapat dilakukan dan sistem akan mengirimkan *response 401 unauthorized*.

Agar penerapan SIAKAD lebih informatif, maka SIAKAD menyediakan *push notification*, yaitu pemberitahuan yang akan muncul disetiap *browser* pengguna. Pemberitahuan yang akan

dimunculkan tentunya pemberitahuan yang terkait dengan proses akademik perguruan tinggi.

3.1.2 Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dikembangkan sebuah perangkat lunak berbasis web yang dapat mengintegrasikan modul-modul yang terdapat pada SIAKAD. Untuk dapat mengintegrasikan modul-modul tersebut maka kerangka kerja yang dibangun bersifat modular dengan menggunakan pola arsitektur HMVC (*Hierarchical Model-View-Controller*) sehingga perubahan, penambahan, dan penghapusan modul dapat dilakukan dengan mudah tanpa mempengaruhi modul-modul lainnya. Modul yang diimplementasikan menggunakan pola arsitektur MVC (*Model-View-Controller*) dengan sedikit memodifikasi arsitekturnya. Kerangka kerja selain untuk mengintegrasikan modul-modul juga untuk mengatur keamanan, *routing*, *logging*, *API*, *RBAC* (*Role Based Access Control*) dan sebagainya seperti pada Gambar 3.2 dibawah ini.



Gambar 3.2 Kerangka kerja SIAKAD

3.1.3 Kebutuhan Fungsional

Berikut merupakan kebutuhan fungsional yang terdapat pada pengerjaan Tugas Akhir ini. Adapun kebutuhan fungsional tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan Fungsional

No	Kode	Fungsional	Deskripsi
1.	F0001	Pengguna dapat mengganti hak akses	Pengguna yang memiliki hak akses lebih dari satu pada SIAKAD, maka pengguna dapat mengganti hak akses
2.	F0002	Mengelola modul pengaturan	Melakukan pengelolaan data pada modul pengaturan yang berupa informasi-informasi yang diperlukan oleh modul-modul lainnya
3.	F0003	Pengguna dapat memilih bahasa pengantar pada sistem	Pengguna dapat memilih bahasa pengantar yang digunakan pada sistem, bahasa yang disediakan adalah bahasa Indonesia dan bahasa Inggris

3.1.4 Kasus Penggunaan

Pada subbab ini akan menunjukkan apa saja yang dapat dilakukan oleh pengguna terhadap sistem. Pada Tugas Akhir ini terdapat pengguna yang merupakan civitas akademik perguruan tinggi yang dikelompokkan menjadi beberapa kelompok pengguna. Tabel 3.2 merupakan tabel kelompok pengguna pada SIAKAD. Gambar 3.3 menjelaskan aktivitas yang dapat dilakukan oleh pengguna terhadap sistem. Untuk kasus penggunaan mengganti hak akses pengguna yang dapat mengganti hak akses adalah pengguna yang memiliki hak akses lebih dari satu.

Pengguna yang memiliki hak akses lebih dari satu adalah pengguna yang merupakan seorang dosen dan memiliki jabatan sebagai kepala satuan manajemen dari suatu unit.

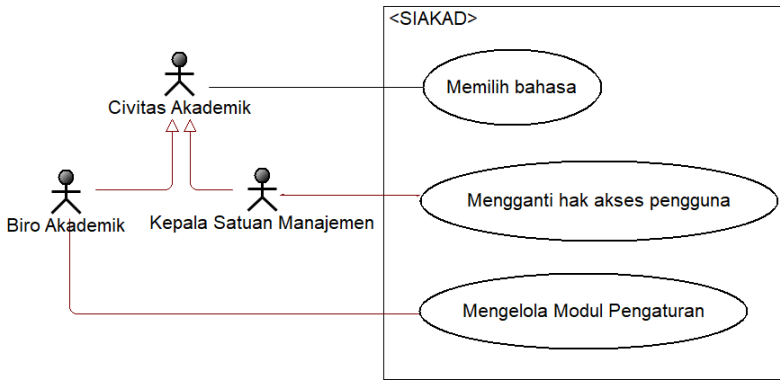
Tabel 3.2 Kelompok pengguna SIAKAD

Kelompok Pengguna	Deskripsi	Contoh
Mahasiswa	Merupakan pengguna yang memiliki hak akses sebagai mahasiswa	Mahasiswa S1 Informatika, mahasiswa S2 Informatika
Dosen	Merupakan pengguna yang memiliki hak akses sebagai dosen	Dosen Informatika, dosen matematika
Tendik	Merupakan pengguna yang memiliki hak akses sebagai tendik	Tata usaha, biro akademik, direktorat akademik.
Kepala Satuan Manajemen	Merupakan pengguna yang memiliki hak akses sebagai ketua dari suatu unit (lab, prodi departemen, fakultas, dsb).	Ketua RMK, kepala prodi, kepala departemen, dekan.

3.1.4.1 Mengelola modul pengaturan

Modul pengaturan dibutuhkan untuk referensi data-data penting yang dibutuhkan oleh modul-modul lainnya. Pada modul pengaturan, terdapat super admin yang dapat mengelola data-data tersebut. data-data tersebut akan disimpan ke dalam sebuah file. Tabel 3.3 merupakan tabel spesifikasi kasus penggunaan untuk

mengelola modul pengaturan dan Gambar 3.4 merupakan diagram aktivitas mengelola modul pengaturan.

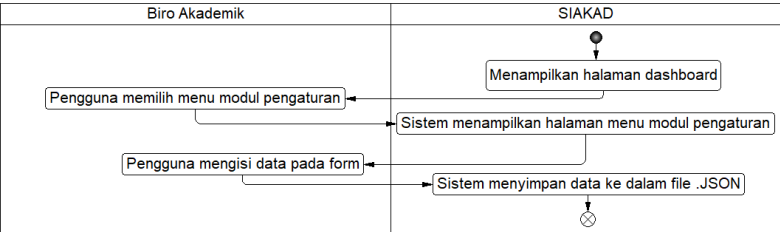


Gambar 3.3 Diagram kasus penggunaan

Tabel 3.3 Spesifikasi kasus penggunaan mengelola modul pengaturan

Komponen	Deskripsi
Nama	Mengelola modul pengaturan
Nomor	UC-001
Deskripsi	Kasus penggunaan ini digunakan untuk mengelola modul pengaturan yang nantinya data-data pada modul pengaturan digunakan pada modul lainnya
Tipe	Fungsional

Komponen	Deskripsi
Aktor	Biro Akademik
Kondisi Awal	Data- data pada modul pengaturan masih kosong ataupun data masih merupakan data lama yang sudah tidak berlaku
Kondisi Akhir	Data pada modul pengaturan merupakan data terbaru
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu pengaturan 2. Sistem menampilkan halaman modul pengaturan 3. Pengguna memasukkan data-data pada form yang telah disediakan pada modul pengaturan 4. Sistem mengubah data-data tersebut ke dalam format JSON 5. Sistem menyimpan data tersebut ke dalam file .json
Alur Alternatif	-



Gambar 3.4 Diagram aktivitas mengelola modul pengaturan

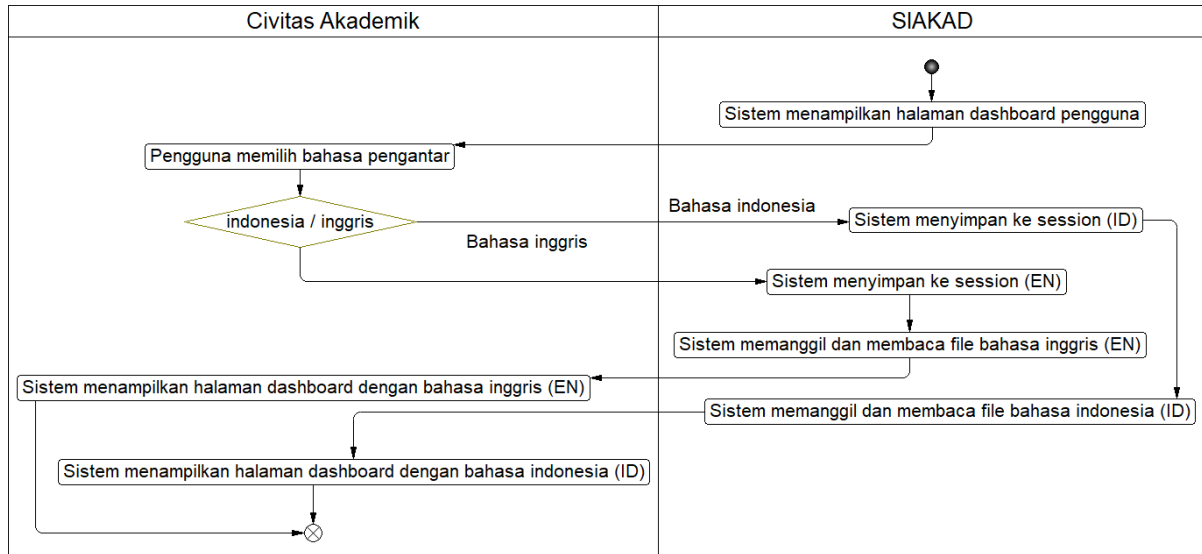
3.1.4.2 Memilih bahasa pengantar

Pengguna pada SIAKAD dapat memilih bahasa pengantar yang digunakan, bahasa pengantar terdiri dari bahasa Indonesia dan bahasa inggris. Tabel 3.4 merupakan tabel spesifikasi kasus penggunaan untuk memilih bahasa pengantar dan Gambar 3.5 merupakan diagram aktivitas memilih bahasa pengantar.

Tabel 3.4 Spesifikasi kasus penggunaan memilih bahasa pengantar

Komponen		Deskripsi
Nama	Memilih bahasa pengantar	
Nomor	UC-002	
Deskripsi	Pengguna dapat mengubah bahasa pengantar pada SIAKAD. Bahasa pengantar yang disediakan adalah bahasa Indonesia dan bahasa inggris	
Tipe	Fungsional	

Komponen	Deskripsi
Aktor	Civitas akademik (semua pengguna SIAKAD yang memiliki hak akses)
Kondisi Awal	Bahasa pengantar pada pertama kali masuk ke SIAKAD merupakan bahasa pengantar yang merupakan bahasa <i>default</i> dipilih pada saat mendaftar akun di myITS.
Kondisi Akhir	Bahasa pengantar berubah sesuai dengan bahasa pengantar yang dipilih oleh pengguna
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu pilihan bahasa yang disediakan 2. Sistem menyimpan pilihan bahasa tersebut ke <i>session</i> 3. Sistem memanggil dan membaca file bahasa yang dipilih oleh pengguna menggunakan <i>session</i> 4. Sistem menampilkan bahasa pengantar ke halaman pengguna sesuai dengan pilihan pengguna
Alur Alternatif	-



Gambar 3.5 Diagram aktivitas memilih bahasa pengantar

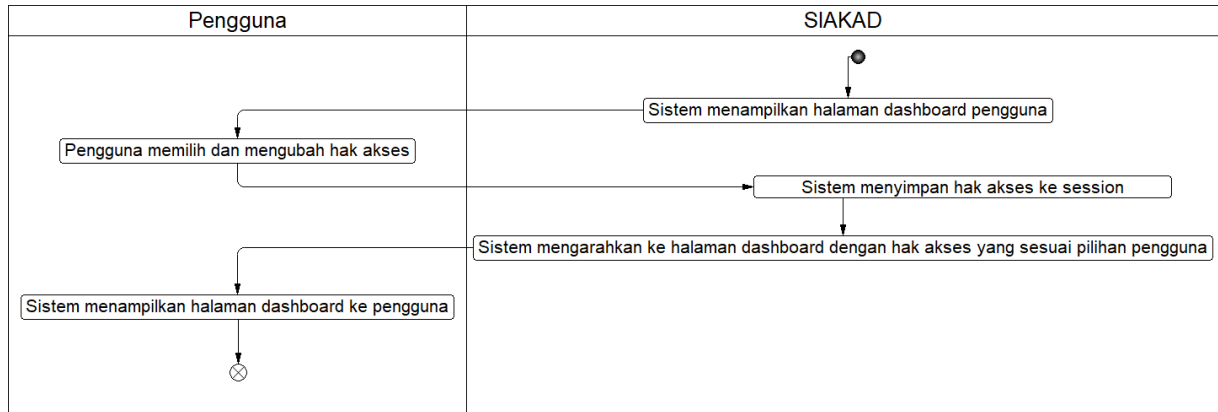
3.1.4.3 Mengganti hak akses pengguna

Pengguna yang memiliki hak akses lebih dari satu maka pengguna tersebut dapat memilih hak akses untuk dapat berpindah dari satu akses ke akses lainnya. Tabel 3.5 merupakan tabel spesifikasi kasus penggunaan untuk mengganti hak akses pengguna dan Gambar 3.6 Gambar 3.5 merupakan diagram aktivitas memilih bahasa pengantar.

Tabel 3.5 Spesifikasi kasus penggunaan mengganti hak akses pengguna

Komponen	Deskripsi
Nama	Mengganti hak akses pengguna
Nomor	UC-003
Deskripsi	Pengguna yang memiliki lebih dari satu hak akses dapat memilih hak akses lainnya untuk dapat akses lainnya
Tipe	Fungsional
Aktor	Kepala satuan manajemen
Kondisi Awal	Akses yang pertama kali pengguna gunakan ketika masuk ke SIAKAD adalah akses <i>default</i> yang ditentukan oleh pengguna pada sistem myITS
Kondisi Akhir	Hak akses pengguna berubah sesuai dengan hak akses yang dipilih oleh pengguna
Alur Normal	1. Pengguna memilih hak akses yang terdapat pada pengguna tersebut

	<ol style="list-style-type: none">2. Sistem menyimpan hak akses pengguna ke <i>session</i>3. Sistem mengarahkan ke halaman <i>dashboard</i> sesuai dengan hak akses yang dipilih oleh pengguna4. Sistem menampilkan halaman <i>dashboard</i> ke pengguna
Alur Alternatif	-



Gambar 3.6 Diagram aktivitas mengganti hak akses pengguna

3.1.5 Kebutuhan Non Fungsional

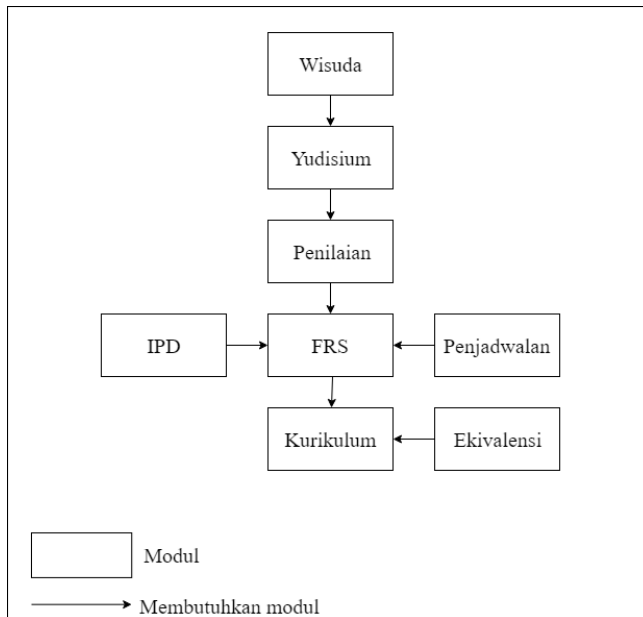
Tabel 3.6 merupakan kebutuhan-kebutuhan non fungsional yang diterapkan pada SIAKAD.

Tabel 3.6 Kebutuhan non fungsional

No	Kode	Non Fungsional
1.	NF001	Sistem menerapkan standar keamanan OAuth2 untuk proses otorisasi
2.	NF002	Sistem menerapkan Role Based Access Control untuk mengatur hak akses pengguna
3.	NF003	Sistem mendukung perubahan penggunaan bahasa pada sistem yaitu penggunaan bahasa Indonesia dan bahasa Inggris
4.	NF004	Sistem menerapkan <i>web push notification</i> untuk memberikan pengumuman kepada pengguna
5.	NF005	Sistem menerapkan API untuk kemudahan pengembangan selanjutnya

3.1.6 Ketergantungan Data Antar Modul

Modul-modul yang akan dibangun memiliki keterkaitan data antara satu modul dengan modul lainnya. Modul-modul tersebut memiliki tingkat ketergantungan seperti Gambar 3.7.



Gambar 3.7 Ketergantungan data antar modul

Gambar 3.7 Ketergantungan data antar modul menunjukkan bahwa modul wisuda membutuhkan data dari modul yudisium, modul yudisium membutuhkan data dari modul penilaian, modul penilaian membutuhkan data dari modul FRS, modul FRS dan ekivalensi membutuhkan data dari modul kurikulum, modul IPD dan penjadwalan membutuhkan data dari modul FRS.

3.2 Perancangan Perangkat Lunak

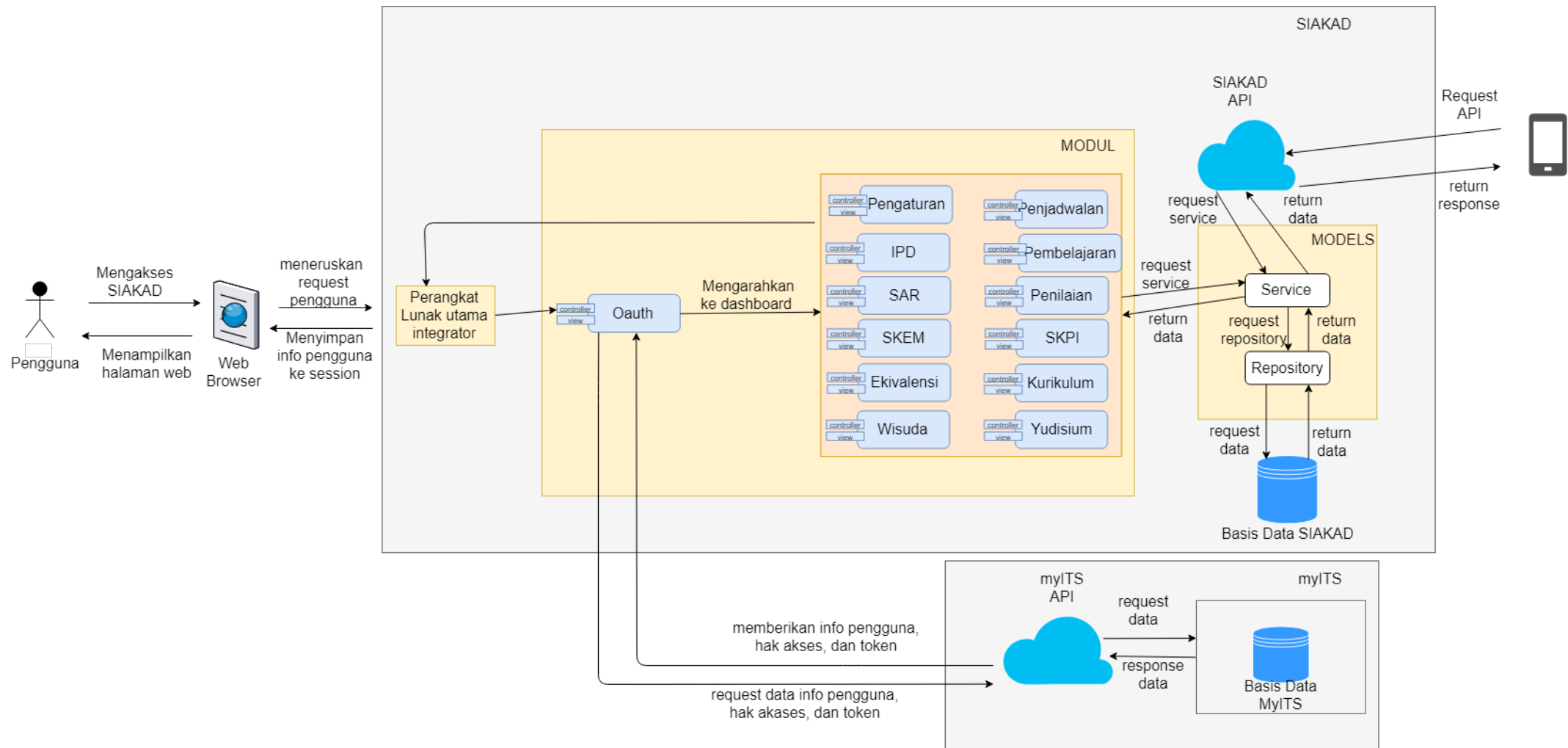
Pada tahap perancangan perangkat lunak, perancangan yang dilakukan yaitu meliputi desain arsitektur yang digunakan untuk modul-modul tersebut yang perlu digabungkan untuk menjadi

suatu sistem secara utuh, dan service-service yang dibutuhkan terhadap sistem.

3.2.1 Desain Arsitektur

Untuk dapat menggabungkan modul-modul tersebut dengan mudah maka diperlukan kerangka kerja yang baik dan memiliki tingkat dependensi (ketergantungan) yang rendah antar modul serta menghindari duplikasi kode antar modul. Modul-modul tersebut akan dibangun menggunakan pola *Model-View-Controller* dengan sedikit memodifikasi pola arsitekturnya yaitu model tidak ditempatkan pada setiap modul akan tetapi ditempatkan secara terpusat, sehingga semua modul mengacu ke satu model, hal ini untuk menghindari duplikasi kode yang berhubungan dengan basis data yang sama yang terdapat pada modul. Permasalahan lain yang akan terjadi diantara lain adalah jika terdapat perbedaan modul-modul yang akan diterapkan pada sistem mungkin akan berbeda antara satu perguruan tinggi dengan perguruan tinggi lainnya, sehingga kerangka kerja yang akan dibangun harus dapat menangani permasalahan tersebut yaitu kerangka kerja harus dapat menambah, mengubah, bahkan menghapus modul dengan mudah.

Untuk dapat menangani permasalahan-permasalahan tersebut maka desain arsitektur yang digunakan pada sistem secara utuh adalah *hierarchical model-view-controller* dan *repository service pattern*. Gambar 3.8 menunjukkan desain arsitektur pada SIAKAD.



Gambar 3.8 Desain arsitektur SIAKAD modular

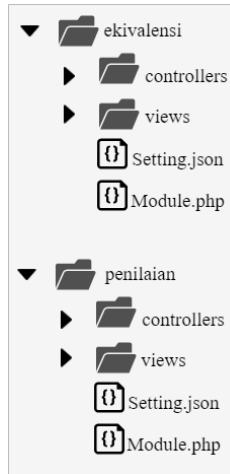
[halaman ini sengaja dikosongkan]

Gambar 3.9 menunjukkan struktur direktori yang akan diimplementasikan oleh modul-modul.



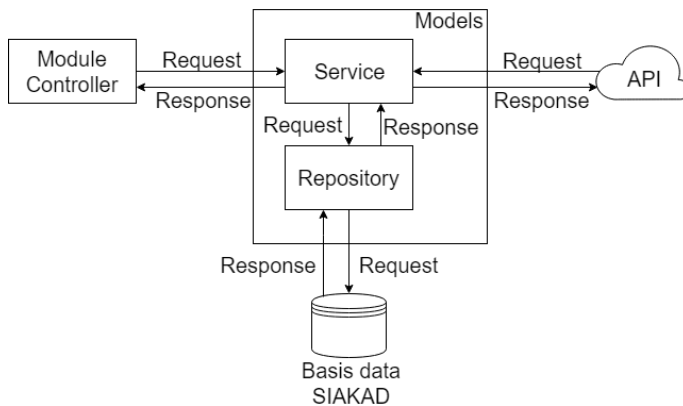
Gambar 3.9 Struktur direktori pada folder modules

Pada Gambar 3.9, setiap modul diimplementasikan ke dalam satu direktori, didalam setiap modul terdapat folder *controller* dan *view*, Gambar 3.10 menunjukkan struktur direktori diterapkan pada setiap modul.



Gambar 3.10 Struktur direktori pada modul

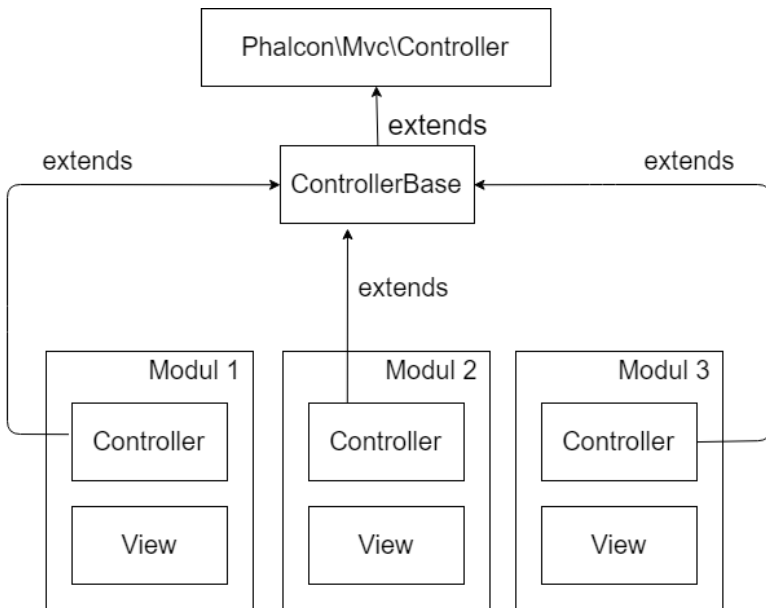
Pada setiap modul diatas, model, yang berguna untuk manajemen basis data, tidak ditempatkan didalam modul. Model ditempatkan pada config sehingga semua model dapat diakses oleh semua modul. Arsitektur yang digunakan pada model adalah *repository service pattern*. Gambar 3.10 merupakan desain arsitektur yang digunakan untuk model. Setiap *controller* pada modul yang diimplementasikan pada SIAKAD akan menggunakan kelas *controllerBase* dimana kelas *controllerBase* akan menggunakan kelas *Phalcon\Mvc\Controller*. Gambar 3.12 menunjukkan diagram hubungan *controller* pada modul dengan kelas *ControllerBase*.



Gambar 3.11 Arsitektur Repository Service Pattern pada SIAKAD

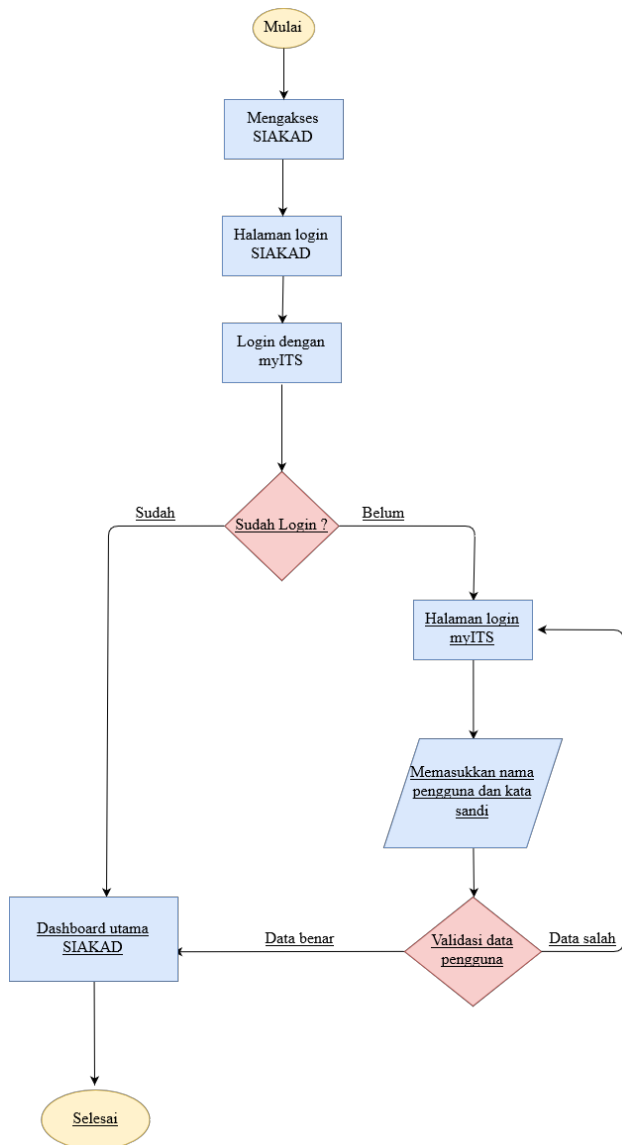
3.2.2 Mekanisme Login

Untuk dapat masuk ke SIAKAD, langkah yang pertama kali harus dilakukan adalah *login*. *Login* adalah suatu proses untuk dapat masuk ke dalam sistem dengan menggunakan informasi rahasia pengguna seperti kata sandi. Untuk dapat *login* ke sistem pengguna harus memasukkan nama pengguna atau id pengguna dan kata sandi pengguna yang sesuai. Pengguna akan melakukan proses otorisasi untuk memberikan hak akses kepada SIAKAD untuk dapat melakukan akses data dari aplikasi myITS. Untuk mekanisme login pada SIAKAD, sistem menerapkan OpenID Connect untuk proses otorisasi dan bertindak sebagai klien dari aplikasi myITS. Aplikasi myITS bertindak sebagai *Authorization Server* (Server Otorisasi). Pada halaman depan SIAKAD, disediakan tombol yang akan mengarahkan pengguna untuk masuk ke sistem menggunakan aplikasi myITS. Pengguna akan diminta untuk memasukkan nama pengguna dan kata sandi yang sesuai, jika data untuk autentikasi sesuai maka selanjutnya aplikasi myITS akan memberikan data pengguna dan melanjutkan request ke halaman dashboard (beranda) SIAKAD. Gambar 3.13 merupakan flowchart yang menggambarkan rangkaian proses login.



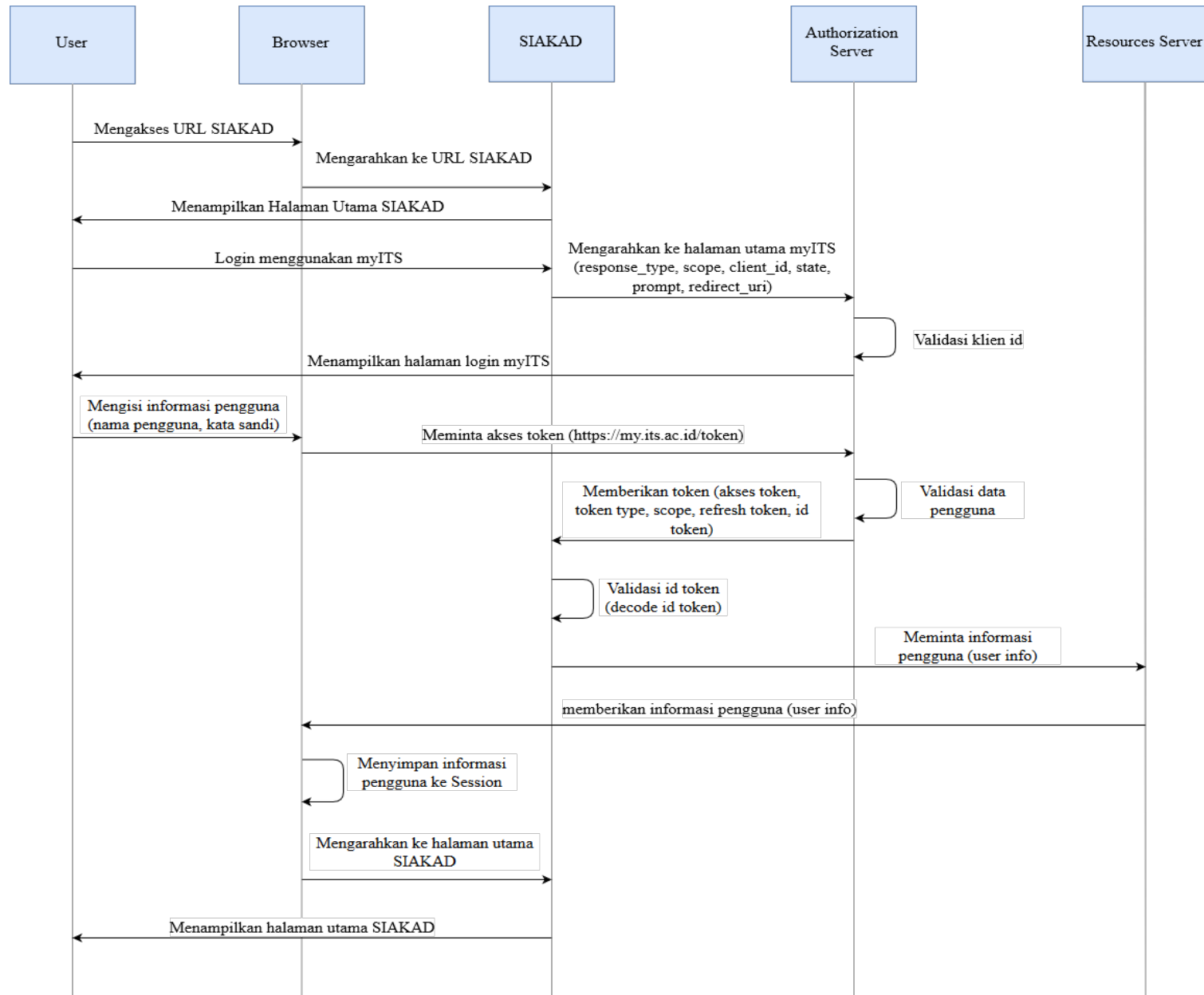
Gambar 3.12 Diagram Controller pada SIAKAD

Adapun mekanisme OAuth2 untuk grant type authorization code flow yang diterapkan pada aplikasi SIAKAD adalah seperti pada Gambar 3.14.



Gambar 3.13 *Flowchart* mekanisme login

[halaman ini sengaja dikosongkan]



Gambar 3.14 *flowchart* authorization code

[halaman ini sengaja dikosongkan]

3.2.3 RBAC

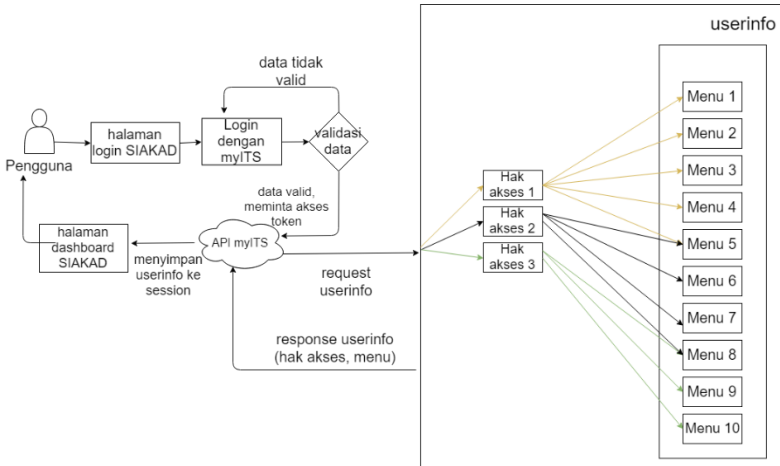
RBAC (*Roled Based Access Control*) diterapkan untuk membatasi hak akses pengguna, hal ini berguna untuk mencegah fitur-fitur dan menu untuk dapat diakses oleh pengguna yang tidak berhak. Masing-masing hak akses pengguna pada SIAKAD memiliki akses terhadap aplikasi yang berbeda-beda. Setiap menu yang terdapat pada SIAKAD di definisikan di aplikasi myITS. Adapun definisi-definisi menu yang didefinisikan pada myITS adalah seperti Gambar 3.15.

Menu		
<u>Menu ID</u>	int	<pk>
Parent ID	int	<fk2>
Client ID	uniqueidentifier	<fk1>
Name Indonesia	varchar(100)	
Name English	varchar(100)	
Description	varchar(255)	
Path	varchar(255)	
Visibility	numeric(1)	
Active	numeric(1)	
Menu Order	numeric(3)	
Icon	varchar(50)	
Create date	datetime	
Last update	datetime	

Gambar 3.15 Spesifikasi tabel menu pada myITS

Setiap menu yang didefinisikan pada myITS memiliki hak akses masing-masing. Dengan adanya penerapan hak akses tersebut maka menu tersebut tidak dapat diakses oleh pengguna yang tidak berhak. Untuk dapat mengakses menu-menu pada SIAKAD, maka pengguna terlebih dahulu harus login ke SIAKAD dengan menggunakan myITS. Jika proses untuk login berhasil, data valid, maka myITS akan mengirimkan informasi ke SIAKAD salah satunya informasi mengenai hak akses dan menu-menu yang dapat diakses oleh pengguna tersebut yang telah disesuaikan dengan hak akses pengguna. Data-data menu tersebut dikirimkan ke SIAKAD menggunakan API dengan bentuk JSON. Menu-menu

tersebut akan diterapkan pada *sidebar* yang terdapat pada halaman *dashboard* pengguna. Pengguna dapat melakukan dan mengakses semua menu yang terdapat pada *dashboard* pengguna. Gambar 3.16 merupakan mekanisme untuk RBAC.



Gambar 3.16 Diagram mekanisme RBAC

3.2.4 Restful API

SIAKAD menyediakan API (*Application Programming Interface*) yang dapat diakses oleh pengguna yang memiliki id token, dimana id token tersebut berupa JWT (JSON Web Token). Pengguna mengirimkan request API dengan mengirimkan header [X-jwt-assertion]. Adapun metode yang dapat digunakan adalah *Post* dan *Get*. Jika token yang dikirimkan valid maka pengguna akan mendapatkan *response* HTTP dengan status 200 dan data yang di *request* akan ditampilkan dalam bentuk JSON ataupun data yang dikirimkan akan disimpan ke basis data SIAKAD. Tabel 3.7 menunjukkan contoh dari perancangan id token JWT.

Tabel 3.7 Contoh JWT yang diterapkan

Header
<pre>{ "typ" : "JWT", "alg" : "RS256" }</pre>
Payload
<pre>{ "iss" : "localhost", "sub" : "0ED7A641-320B-49F5-9944-C5A4478B168E", "aud" : "A7600479-013E-4377-A320-602281F8D286", "iat" : 1525222046, "exp" : 1525225646, "auth_time" : 1525222046 }</pre>
Verify Signature
<p>RSASHA256 (</p> <p>Base64UrlEncode(header) + "." + base64UrlEncode(payload),</p>

<pre> -----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCg KCAQEA0dKg98uts1FzL1MMAS/Prbi38lYy7ofseDte NJrjsyPUQTcbp/S+E0YeVhazqo8iKfiB1ItjVo1CLcuQI d2De4aitl7MAvSfvSUpe/nj5MZgetyOPemt/hqx7QVM dk98X6ihC4GktXBZNptf4L2fSxxu3sptl8NuZXfHGGH uyFyTpQukpR9h4X50VKzLGGpARxVY50bBaMpEjF sAoNsbuBntBwFfvVIBhBn4uPXjbp0XpB/SjDmkAOy 0HckUs/iBjmCNTbh0xLbP/mNABQVQ18jvnhYK7pm q/4/E1VEwI4IMwNY347NfUOn+t9IW9mpoyvQx2GC hYyR3VPYPnoLfQwIDAQAB -----END PUBLIC KEY----- </pre>

Tabel 3.7 menunjukkan desain dari perancangan JWT yang nantinya akan di *encode* menggunakan *library* firebase JWT. Format pada token JWT terdiri dari *header*, *payload*, dan *signature*. *Header* menunjukkan informasi tipe dan algoritma yang digunakan, dalam contoh tabel diatas menggunakan RS256, pada SIAKAD algoritma RS256 juga dipakai untuk melakukan decode id token nantinya. Adapun informasi pada *payload* terdiri dari *iss* yang menunjukkan *issuer* dari token, *sub* menunjukkan *subject* (id pengguna) dari token, *aud* menunjukkan *audience* dari token (id client), *iat* menunjukkan waktu JWT dikeluarkan dan *exp* menunjukkan waktu kadaluarsa dari token tersebut. Tabel 3.8 menunjukkan informasi JWT yang sudah di decode.

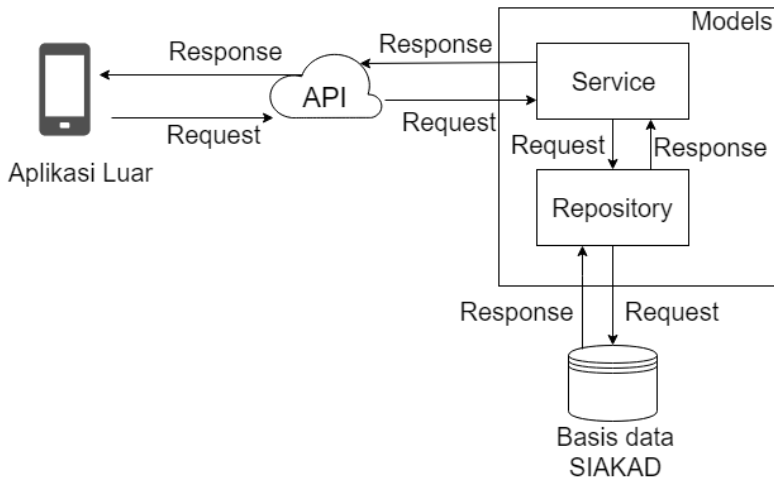
Tabel 3.8 JWT yang telah di decode

```

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJs
b2NhbGhvc3QiLCJzdWIiOiIwRUQ3QTY0MS0zMjBCLTQ
5RjUtOTk0NC1DNUE0NDc4QjE2OEUiLCJhdWQiOiJBZnZ
YwMDQ3OS0wMTNFLTQzNzctQTM5MC02MDIyODFG
OEQyODYiLCJpYXQiOiE1MjUyMjIwNDYsImV4cCI6M
TUyNTIyNTY0NiwiYXV0aF90aW1lIjoxNTI1MjIyMDQ2f
Q.N3b2ZJF8SJcBpgP_xZoGlPG8bV97WFaMu6z1V_fEcC6
Zc3UUDjuqJHJUZsp-
7lvPoLUd819pUyuVci3JuCYOG0Cx9by8ps2bdezG4JRy3-
QeNdyN_WKkRNrH4Qr_7ihjuFxIInRhICrmO5h9xA4TUyf
EN8niDFYvOnuRc_Auyhwb7VIBppuiIbCO-
ejr7xePNRRimFugA6V2rGjBEEORcL46bOWf_-bhFkHG-
XtlSaD5d0xQwEwwGkN857CjxLP7xYHirHy5UsmLbiKFY
2PpGGIr8XrvuqeXqsKtNvZ4RKh9_cqzu8GkBk7aEUOxFZ

```

Informasi tersebut di *decode* menggunakan algoritma RS256. Terdapat 3 bagian yang dipisahkan oleh titik pada setiap bagiannya. Bagian yang pertama, yang berwarna merah, merupakan informasi dari *header*. Bagian yang kedua, yang berwarna ungu, merupakan informasi *payload* dan bagian yang terakhir adalah *signature key*, dimana pada bagian tersebut berisikan *public key* yang selanjutnya harus di validasi menggunakan *private key*. Adapun mekanisme untuk mengakses API adalah seperti Gambar 3.17.



Gambar 3.17 Mekanisme API pada SIAKAD

3.2.5 Routing

Routing pada SIAKAD digunakan untuk menentukan modul, *controller*, dan *action* yang sesuai dengan permintaan dari pengguna. Adapun perancangan untuk routing adalah sebagai berikut.

1. Penamaan *Controller* menggunakan format StudlyCase, contoh: LihatNilaiMahasiswaController
2. Penamaan *view* menggunakan snake_case, contoh: data_mahasiswa

Format routing yang diterapkan pada SIAKAD adalah “Alamat IP/Modul/Controller”, dimana alamat IP (*Internet Protocol*) merupakan IP ataupun nama domain dari aplikasi SIAKAD, modul merupakan modul yang diakses oleh pengguna, dan controller yang akan menangani permintaan tersebut. Nama *controller* yang menggunakan format StudlyCase akan diterjemahkan menjadi format snake_case pada saat pengguna mengakses. Tabel 3.9 berikut merupakan contoh dari alamat IP, modul, dan *controller*.

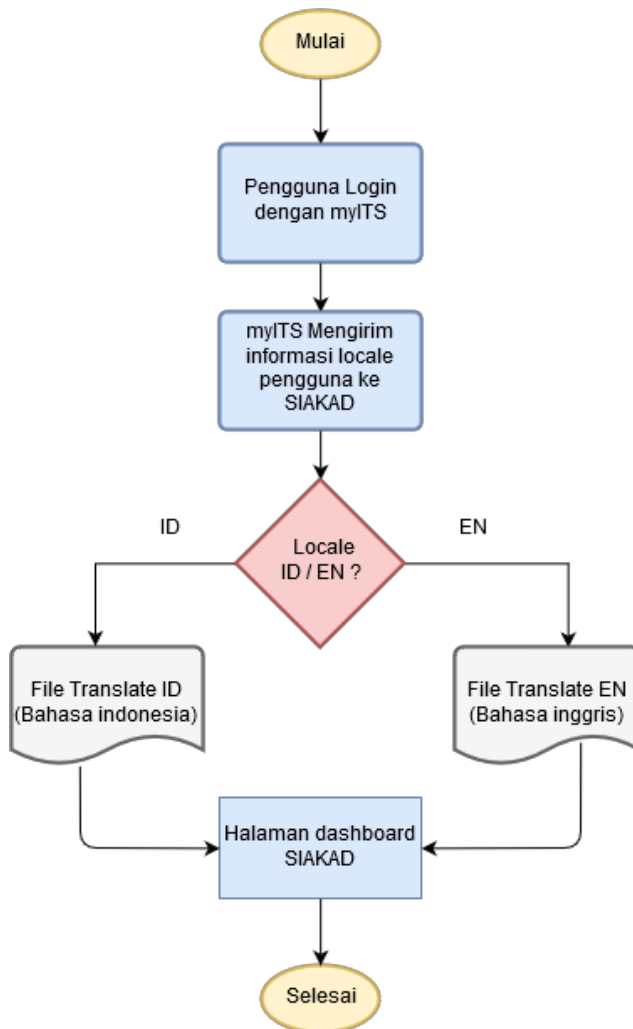
Tabel 3.9 Parameter format routing

Alamat IP/Domain	Modul	Controller
127.0.0.1	Penilaian	LihatNilaiMahasiswaController

Untuk dapat mengakses halaman tersebut maka routing yang digunakan adalah 127.0.0.1/Penilaian/lihat-nilai-mahasiswa.

3.2.6 Internationalization (i18n)

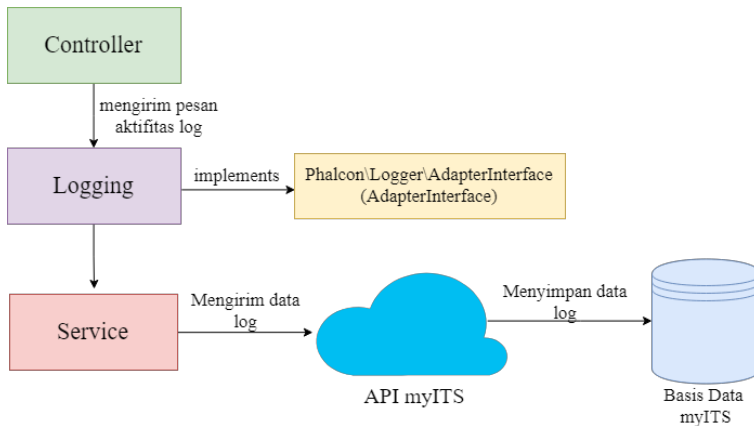
Internationalization atau biasa dikenal dengan istilah i18n adalah proses merancang aplikasi perangkat lunak sehingga dapat diadaptasi ke berbagai bahasa dan wilayah. SIAKAD menyediakan dua pilihan bahasa yaitu bahasa Indonesia dan bahasa Inggris sebagai bahasa pengantar. Pada saat pertama kali masuk ke dalam SIAKAD, sistem akan menampilkan bahasa pengantar ke pengguna sesuai dengan informasi yang didapatkan pada myITS. myITS akan memberikan informasi ke SIAKAD, salah satunya adalah informasi mengenai asal pengguna. Informasi tersebut adalah *locale* yang berisikan nilai ID ataupun EN, dimana ID merupakan bahasa Indonesia sebagai bahasa pengantar yang digunakan pada SIAKAD dan EN merupakan bahasa Inggris sebagai bahasa pengantar yang digunakan. Pengguna juga dapat mengganti bahasa pengantar yang digunakan pada SIAKAD dengan memilih opsi yang disediakan pada *topbar* SIAKAD. Gambar 3.18 menunjukkan *flowchart* untuk mekanisme bahasa pengantar yang akan digunakan yang sesuai dengan informasi dari *locale* pengguna pada myITS.



Gambar 3.18 *Flowchart i18n*

3.2.7 Logging

SIAKAD akan mencatat setiap aktivitas (log) yang dilakukan oleh pengguna terhadap sistem. Setiap log akan dikategorikan kedalam beberapa kategori yaitu *debug*, *error*, *info*, *notice*, *warning*, *alert*, dan *emergency*. Log pada aplikasi SIAKAD akan disimpan pada basis data milik myITS. myITS menyediakan API untuk client nya salah satunya adalah API untuk memasukkan data log yang diterima dari aplikasi klien. Gambar 3.19 merupakan mekanisme untuk mencatat log pada aplikasi SIAKAD ke myITS.



Gambar 3.19 Mekanisme logging

Pada Gambar 3.19, *controller* akan memanggil *logging* untuk mengirim pesan aktivitas log yang dilakukan pengguna, setiap kali *action* pada *controller* tersebut dijalankan maka *controller* akan memanggil *logging* dan *logging* akan menerima pesan yang dikirimkan oleh *controller*. *Logging* merupakan kelas yang mengimplementasikan kelas *interface* yaitu *Phalcon\Logger\AdapterInterface*. Pada kelas *logging*, terdapat data-data lainnya selain data log yang berasal dari *controller*, data-

data tersebut adalah `client_id`, `user_id`, `category`, `activity`, `time`, `ip address`, `hostname`, `os`, dan `browser`. Data-data tersebut selanjutnya akan dikirimkan ke API log myITS dan selanjutnya akan disimpan ke dalam basis data myITS. Gambar 3.20 merupakan perancangan desain dari tabel log yang terdapat pada basis data myITS.

Activity Log		
<u>Log ID</u>	<u>bigint</u>	<u><pk></u>
Client ID	uniqueidentifier	<fk2>
User ID	uniqueidentifier	<fk1>
Category	numeric	
Activity	varchar(255)	
Time	datetime	
IP address	<Undefined>	
Hostname	<Undefined>	
OS	varchar(100)	
Browser	varchar(100)	

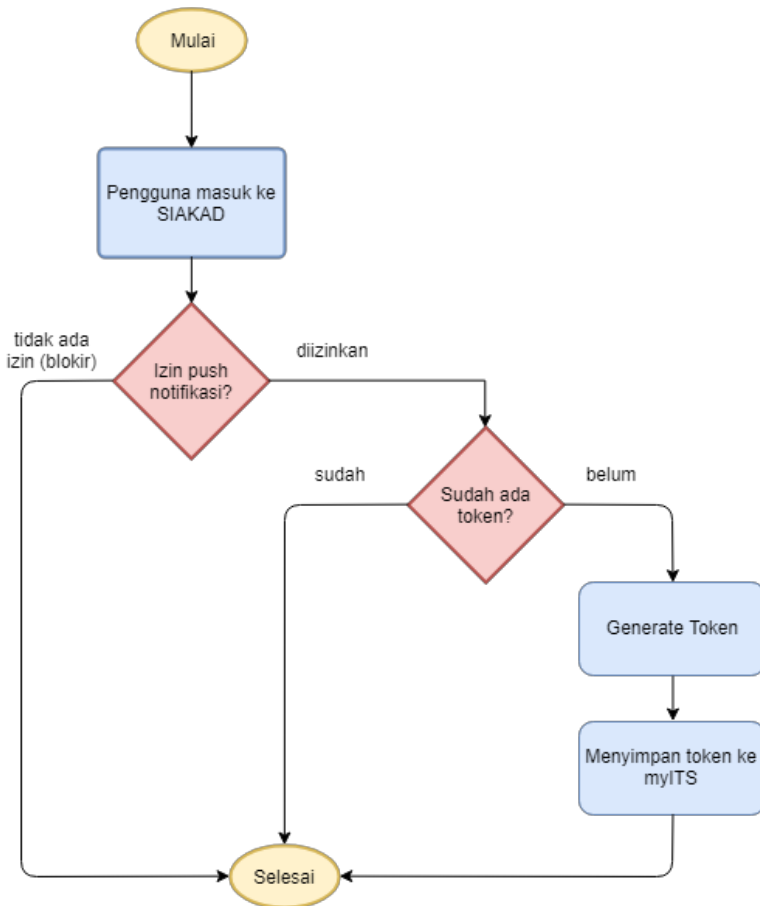
Gambar 3.20 Spesifikasi tabel log pada myITS

3.2.8 Push Notification

SIKAD menyediakan fitur untuk dapat memberitahukan pemberitahuan penting kepada pengguna. Pemberitahuan ini dapat disampaikan melalui fitur *push notification*. Pengguna yang mengizinkan SIKAD untuk dapat menerima pemberitahuan melalui *push notification* akan mendapatkan suatu informasi apabila ada informasi terbaru, informasi akan disajikan kedalam bentuk *pop-up* yang akan tampil ketika pesan disampaikan. Pengguna yang tidak mengizinkan fitur *push notification* maka tidak dapat menerima pemberitahuan berupa *pop-up*. Fitur *push notification* ini dapat di izinkan dan dapat di blokir (tidak diizinkan), apabila fitur ini diizinkan oleh pengguna, maka SIKAD akan mengirimkan sebuah token ke basis data myITS untuk disimpan, token tersebut merupakan sebuah informasi

mengenai perangkat yang digunakan oleh pengguna sehingga apabila ada pesan yang disampaikan maka pesan tersebut dapat disampaikan ke pengguna yang benar.

Fitur *push notification* ini menggunakan firebase cloud messaging untuk web push notification Gambar 3.21 merupakan *flowchart* yang menunjukkan apabila pengguna mengizinkan dan tidak mengizinkan fitur *push notification*.



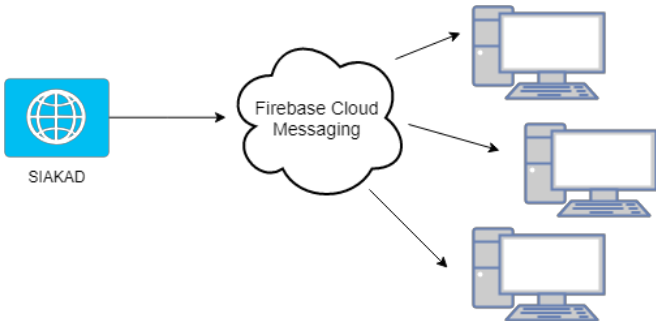
Gambar 3.21 *flowchart web push notification untuk device token*

Token tersebut akan disimpan ke basis data myITS yang memiliki struktur seperti Gambar 3.22.

Device Token		
Device token ID	uniqueidentifier	<pk>
Client ID	uniqueidentifier	<fk2>
User ID	uniqueidentifier	<fk1>
Device token	varchar(255)	
Device type	char(1)	
Active	numeric(1)	
Timestamp	datetime	

Gambar 3.22 Spesifikasi tabel device token pada myITS

Untuk dapat mengirimkan sebuah pesan, maka SIAKAD memanfaatkan firebase cloud messaging. Pada kasus ini, SIAKAD akan mengirimkan pesan ke API yang disediakan oleh firebase cloud messaging dan pesan akan diteruskan ke pengguna yang mengizinkan fitur *push notification*. Gambar 3.23 merupakan diagram yang menggambarkan proses pengiriman pesan.



Gambar 3.23 Mekanisme pengiriman pesan pada fitur *push notification*

3.2.9 Konvensi Kode

Konvensi kode pada SIAKAD yaitu digunakan untuk mengatur standar penulisan kode pada PHP, HTML, dan JavaScript. Pada PHP standar penulisan kode yang digunakan yaitu mengikuti standar PSR-1 dan PSR-2, penulisan HTML dan JavaScript mengikuti standar aturan W3Schools. Adapun contoh penulisan kode seperti dibawah ini.

a. PHP

1. Penulisan nama kelas menggunakan penulisan StudlyCaps, dimana huruf awal setiap kata harus menggunakan huruf kapital, dan setiap kata di sambung tanpa spasi ataupun menggunakan karakter lain seperti *underscore*. Contoh: `AuthorizeControllor`.
2. Penulisan nama *method* atau *action* menggunakan penulisan camelCase, dimana huruf pada kata pertama menggunakan huruf kecil dan huruf pada kata kedua dan selanjutnya menggunakan huruf kapital.
Contoh: `getAuthorizationResponseAction()`.
3. Indentasi penulisan kode menggunakan 4 spasi (1 tab pada *keyboard*).
4. Penggunaan kurung kurawal (*bracket*) pembuka pada nama kelas dan method digunakan dibawah nama kelas atau dibaris setelah nama kelas, sedangkan penggunaan kurung kurawal pada kondisi (`for`, `if`, `elseif`, `while`) digunakan disamping nama kondisi tersebut dan penggunaan kurung kurawal penutup digunakan dibaris selanjutnya.

Gambar 3.24 merupakan contoh penerapan konvensi kode pada PHP.

```

1  <?php
2  namespace Vendor\Package;
3  // baris kosong
4  use FooInterface;
5  use BarClass as Bar;
6  use OtherVendor\OtherPackage\BazClass;
7  // baris kosong
8  class Foo extends Bar implements FooInterface
9  {
10     public function sampleMethod($a, $b = null)
11     {
12         if ($a === $b) {
13             bar();
14         } elseif ($a > $b) {
15             $foo->bar($arg1);
16         } else {
17             BazClass::bar($arg2, $arg3);
18         }
19     }
20
21     public function anotherMethod($expr)
22     {
23         while ($expr) {
24             for ($i = 0; $i < 10; $i++) {
25                 // for body
26             }
27         }
28     }
29
30     public final lastMethod()
31     {
32         try {
33             // try body
34         } catch (FirstExceptionType $e) {
35             // catch body
36         }
37     }
38 }

```

Gambar 3.24 Penulisan kode PHP menggunakan PSR-1 dan PSR-2

b. HTML

1. Penulisan elemen pada HTML menggunakan huruf kecil. Contoh:

```

<section>
  <p>This is a paragraph.</p>
</section>

```

2. Setiap elemen harus memiliki penutup elemen, Contoh: <p>this is paragraph</p>

3. Penulisan atribut pada HTML menggunakan huruf kecil, Contoh: `<div class="menu">`
4. Penulisan *values* pada atribut menggunakan tanda petik dua (quotes), Contoh: `<table class="striped">`
5. Penulisan elemen `` selalu diberikan atribut alt, hal ini berguna disaat gambar tidak dapat di muat pada halaman web. Contoh:
``
6. Penulisan tanda “=” antara atribut dan values tidak dipisahkan oleh spasi. Contoh:
`<link rel="stylesheet" href="styles.css">`
7. Menghilangkan `<html>` dan `<body>` tidak direkomendasikan

c. JavaScript

1. Penulisan nama variabel menggunakan penulisan camelCase, contoh: `firstName`.
2. Menggunakan spasi diantara setiap operator, contoh:
`var x = y + z;`
3. Indentasi penulisan kode menggunakan 4 spasi (1 tab pada *keyboard*).
4. Penulisan kurung kurawal pembuka pada *statements* (function, loops, conditionals) berada disamping *statements* dan kurung kurawal penutup berada dibawah, contoh seperti Gambar 3.25.

```

1
2 // function
3 function toCelsius(fahrenheit) {
4     return (5 / 9) * (fahrenheit - 32);
5 }
6
7 // loops
8 for (i = 0; i < 5; i++) {
9     x += i;
10 }
11
12 //conditionals
13 if (time < 20) {
14     greeting = "Good morning";
15 } else {
16     greeting = "Good evening";
17 }
18

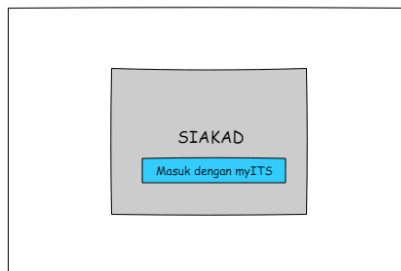
```

Gambar 3.25 Konvensi kode JavaScript

3.2.10 Perancangan Antar Muka

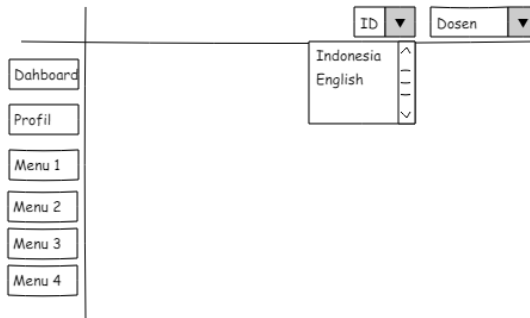
Perancangan antar muka dilakukan terhadap rancangan halaman utama SIAKAD (halaman login), Perancangan opsi untuk memilih bahasa, perancangan untuk memilih opsi hak akses, dan perancangan halaman melihat log dari aplikasi. Berikut merupakan gambar-gambar dari perancangan antar muka.

1. Perancangan antar muka untuk halaman login



Gambar 3.26 Perancangan antar muka untuk halaman login

2. Perancangan antar muka untuk memilih bahasa



Gambar 3.27 Perancangan antar muka untuk memilih bahasa

3. Perancangan antar muka untuk memilih opsi hak akses



Gambar 3.28 Perancangan antar muka untuk memilih opsi hak akses

4. Perancangan antar muka untuk 404 not found

The wireframe shows a web application layout. At the top right, there are two dropdown menus labeled 'Id' and 'Admin'. On the left side, there is a vertical menu with five items: 'Dashboard', 'Profil', 'Pengaturan' (highlighted in cyan), 'Menu 2', 'Menu 3', and 'Menu 4'. The main content area is divided into two sections. The top section is a box containing three lines of text: 'Informasi 1', 'Informasi 2', and 'Informasi 3'. The bottom section is a form with four labels: 'Pengaturan 1', 'Pengaturan 2', 'Pengaturan 3', and 'Pengaturan 4', each followed by a text input field. At the bottom right of this form is a green button labeled 'simpan'.

Gambar 3.29 Perancangan antar muka 404 not found

5. Perancangan antar muka untuk modul pengaturan



Gambar 3.30 Perancangan antar muka untuk modul pengaturan

BAB IV IMPLEMENTASI

Pada bab ini akan dijelaskan tahap-tahap implementasi dari perangkat lunak utama berdasarkan perancangan yang telah dibahas pada bab sebelumnya dan aturan pembuatan modul perangkat lunak Sistem Informasi Akademik (SIKAD). Penjelasan aturan pembuatan modul akan disertai langkah-langkah pembuatan modul. Selain itu juga akan dibahas mengenai persiapan lingkungan pengembangan, langkah-langkah pengembangan perangkat lunak web dengan framework Phalcon.

Pada bab ini, implementasi akan dijelaskan berdasarkan lingkup utama dari perangkat lunak, yaitu modularitas, akses data, *template* antarmuka pengguna, dan keamanan, dan fitur-fitur pendukung lainnya.

4.1 Lingkungan Implementasi

Implementasi dilakukan pada dua lingkungan yang berbeda, yaitu lingkungan untuk pengembangan (localhost) dan lingkungan untuk produksi (server). Tabel 3.1 merupakan implementasi perangkat lunak pada lingkungan pengembangan.

Tabel 4.1 Spesifikasi lingkungan pengembangan

Spesifikasi	Deskripsi
Sistem Operasi	Windows 10 Education 64 bit
Bahasa pemrograman	PHP versi 7.0
Framework	Phalcon versi 3.3
Basis data	Sql Server 2017 Express Edition
Web server	Apache Web Server

Adapun lingkungan implementai pada lingkungan produksi adalah seperti pada Tabel 4.2.

Tabel 4.2 Spesifikasi lingkungan produksi

Spesifikasi	Deskripsi
Sistem Operasi	Debian 9
Bahasa pemrograman	PHP versi 7.0
Framework	Phalcon versi 3.3
Basis data	Sql Server 2017 Express Edition
Web server	Nginx Web Server

4.2 Implementasi Modularitas

Untuk dapat membangun SIAKAD modular maka SIAKAD akan dibangun dengan pola arsitektur *Hierarchical Model View Controller (HMVC)*. Penambahan, penghapusan, pengubahan pada modul menjadi lebih mudah, dan untuk mempelajari sistem dapat dipecah menjadi beberapa modul.

Pada SIAKAD, untuk dapat menambahkan modul maka hal yang harus dilakukan adalah mendaftarkan modul tersebut di folder config pada *file* modules.php dan loader.php. Kode sumber 4.1 merupakan implementasi yang dilakukan untuk mendaftarkan modul-modul SIAKAD pada loader.php.

```

1. 'Siakad\Oauth' => APP_PATH . '/modules/oauth',
2. 'Siakad\Api' => APP_PATH . '/modules/api/v1',
3. 'Siakad\Api2' => APP_PATH . '/modules/api/v2',
4. 'Siakad\Dashboard' => APP_PATH . '/modules/dashboard',
5. 'Siakad\Ekivalensi' => APP_PATH . '/modules/login',
6. 'Siakad\Ipd' => APP_PATH . '/modules/ipd',
7. 'Siakad\Kurikulum' => APP_PATH . '/modules/kurikulum',
8. 'Siakad\Frs' => APP_PATH . '/modules/frs',
9. 'Siakad\Penilaian' => APP_PATH . '/modules/penilaian',
10. 'Siakad\Penjadwalan' => APP_PATH . '/modules/penjadwalan',
11. 'Siakad\Periode' => APP_PATH . '/modules/periode',

```

```

12. 'Siakad\Ref' => APP_PATH . '/modules/ref',
13. 'Siakad\Sar' => APP_PATH . '/modules/sar',
14. 'Siakad\Skem' => APP_PATH . '/modules/skem',
15. 'Siakad\Skpi' => APP_PATH . '/modules/skpi',
16. 'Siakad\User' => APP_PATH . '/modules/user',
17. 'Siakad\Wisuda' => APP_PATH . '/modules/wisuda',
18. 'Siakad\Yudisium' => APP_PATH . '/modules/yudisium',
19. 'Siakad\Pengaturan' => APP_PATH . '/modules/pengaturan',

```

Kode sumber 4.1 mendaftarkan modul pada loader.php

Adapun format yang digunakan untuk pemberian namespace pada setiap modul di loader.php adalah 'SIKAD\Nama Modul'. Setelah semua modul didefinisikan pada loader.php, maka selanjutnya mendefinisikan file Module.php setiap modul pada file modules.php, Kode sumber 4.2 merupakan contoh untuk mendefinisikan Module.php pada modul penilaian di modules.php.

```

1. 'penilaian' => [
2.     'className' => 'Siakad\Penilaian\Module',
3.     'path'      => APP_PATH . '/modules/penilaian/Module.php'
4. ]

```

Kode sumber 4.2 mendaftarkan module.php pada modules.php

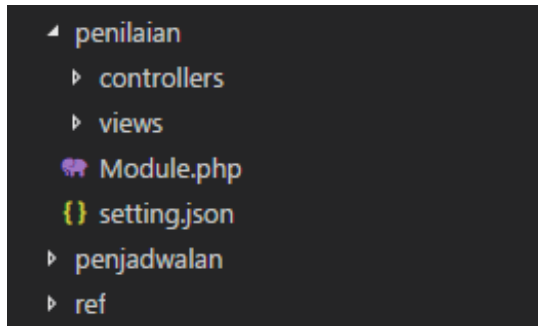
Untuk mendaftarkan setiap modul pada modules.php maka kriteria yang dibutuhkan adalah mendefinisikan className, dimana className akan digunakan sebagai namespace pada file module.php. Format untuk penamaan className adalah seperti pada Tabel 4.3.

Tabel 4.3 Kriteria ClassName

Nama Aplikasi	Modul	Class
Siakad	Penilaian	Module

Kriteria selanjutnya adalah path yang mendefinisikan lokasi dari file module.php pada modul tersebut, dalam contoh ini adalah modul penilaian. Format untuk mendefinisikan path adalah `APP_PATH . 'modules/nama modul/Module.php'`.

Setiap modul pada SIAKAD mempunyai file module.php yang berguna untuk mengatur kebutuhan modul tersebut. *Controller* dan *views* yang terdapat pada setiap modul akan didefinisikan pada module.php dan penggunaan format `view .volt` sebagai engine untuk *views* juga akan didefinisikan di module.php. Gambar 4.1 merupakan struktur yang terdapat pada modul penilaian.



Gambar 4.1 struktur modul penilaian

Terdapat file `Module.php` pada folder penilaian. *File Module.php* akan mendefinisikan kebutuhan-kebutuhan *service* pada modul penilaian, Kode sumber 4.3 menunjukkan cara untuk mendefinisikan *controllers* dan *views* yang dimiliki oleh modul penilaian pada file `Module.php`.

```

1. public function registerAutoloaders(DiInterface $di = null)
2. {
3.     $loader = new Loader();
4.
5.     $loader->registerNamespaces([
6.         'Siakad\Penilaian\Controllers' => __DIR__ . '/controllers/',
7.         'Siakad\Models\Repositories' => APP_PATH . '/models/repositories'
8.         '/',
9.         'Siakad\Models\Services' => __DIR__ . '/../models/services/'
10.    ]);
11.    $loader->register();
12. }

```

Kode sumber 4.3 mendefinisikan controllers dan views module

Adapun Definisi-definisi yang diterapkan pada SIAKAD yang terdapat pada setiap Module.php pada setiap modul adalah seperti berikut.

1. Definisi lokasi controllers dan namespace untuk controllers pada modul tersebut.
2. Definisi lokasi views dan namespace untuk views pada modul tersebut.
3. Definisi basis data yang digunakan pada modul tersebut.
4. Definisi *service* yang digunakan pada modul tersebut (seperti penggunaan .volt untuk file halaman view).

Untuk mendefinisikan Controller pada setiap modul agar dapat di akses melalui route, yang harus dilakukan adalah memberikan namespace pada setiap controller. Kode sumber 4.4 menunjukkan penggunaan namespace pada controller modul penilaian.

```

1. <?php
2. namespace Siakad\Penilaian\Controllers;
3.
4. use Siakad\Models\Services\Services;
5. use Siakad\Controller\ControllerBase;

```

```

6.
7. class DataEvaluasiController extends ControllerBase{}

```

Kode sumber 4.4 namespace pada controller penilaian

Kode sumber 4.5 menunjukkan penggunaan namespace pada controller modul sar.

```

1. <?php
2. namespace Siakad\Penilaian\Controllers;
3.
4. use Siakad\Models\Services\Services;
5. use Siakad\Controller\ControllerBase;
6.
7. class DataEvaluasiController extends ControllerBase{}

```

Kode sumber 4.5 namespace pada controller sar

Pada dua contoh controller diatas (Kode sumber 4.4 dan Kode sumber 4.5), penulisan namespace pada controller harus sesuai dengan namespace yang didefinisikan pada file Module.php pada modul tersebut. Pada Kode sumber 4.5 untuk mendefinisikan controller pada modul penilaian adalah 'Siakad\Penilaian\Controller', maka setiap file controller pada modul penilaian harus menggunakan namespace yang sama yaitu 'Siakad\Penilaian\Controller'. Hal tersebut juga berlaku pada modul SAR dan modul-modul lainnya yang terdapat pada SIAKAD.

Setiap *controller* pada semua modul *extends* class ControllerBase, dimana ControllerBase adalah *controller* yang ditempatkan secara terpusat, dan definisi-definisi umum yang digunakan *controller* pada setiap modul akan didefinisikan dan di terapkan pada class ControllerBase. File ControllerBase *extends* class *controller* Phalcon agar semua *controller* dapat dikenali dan diakses dengan route. Penerapan *controller* pada SIAKAD seperti pada Gambar 3.12.

Untuk dapat masuk ke SIAKAD, langkah yang pertama kali harus dilakukan adalah *login*. *Login* adalah suatu proses untuk dapat masuk ke dalam sistem dengan menggunakan informasi rahasia pengguna seperti kata sandi. Untuk dapat login ke sistem pengguna harus memasukkan nama pengguna atau id pengguna dan kata sandi pengguna yang sesuai. Pengguna akan melakukan proses otorisasi untuk memberikan hak akses kepada SIAKAD untuk dapat melakukan akses data dari aplikasi myITS. Untuk mekanisme *login* pada SIAKAD, sistem menerapkan OpenID Connect untuk proses otorisasi dan bertindak sebagai klien dari aplikasi myITS. Aplikasi myITS bertindak sebagai Authorization Server (Server Otorisasi). Pada halaman depan SIAKAD, disediakan tombol yang akan mengarahkan pengguna untuk masuk ke sistem menggunakan aplikasi myITS. Pengguna akan diminta untuk memasukkan nama pengguna dan kata sandi yang sesuai, jika data untuk autentikasi sesuai maka selanjutnya aplikasi myITS akan memberikan data pengguna dan melanjutkan *request* ke halaman *dashboard* (beranda) SIAKAD.

Implementasi pada Kode sumber 4.4 dan Kode sumber 4.5 *controller* tersebut extends *ControllerBase* dimana kelas *ControllerBase* tersebut seperti pada Kode sumber 4.6.

```

1. <?php
2. namespace Siakad\Controller;
3.
4. use Phalcon\Mvc\Controller;
5. use Phalcon\Translate\Adapter\NativeArray;
6. use Phalcon\Logger\Adapter\Logging as Log;
7. use Siakad\Models\Services\Services;
8. use Phalcon\Http\Response;
9.
10. class ControllerBase extends Controller
11. {
12.     protected $data;
13.     protected $logger;
14.     protected $state;

```

```

15.   protected $can_insert;
16.   protected $can_update;
17.   protected $can_delete;
18.   protected $setting;
19.
20.   public function onConstruct()
21.   {
22.       if($this->session->get('user')!=null){
23.           $this->logger = new Log();
24.           $this->checkAccessControl();
25.           $this->setting = json_decode($this->getFilePengaturan());
26.       }else{
27.           header("Location: /oauth");
28.       }
29.   }
30. }

```

Kode sumber 4.6 file ControllerBase

Pada Kode sumber 4.6 memiliki sebuah fungsi yang berfungsi sebagai konstruktor. Setiap kali *controller* pada modul dipanggil maka konstruktor tersebut akan dieksekusi. Konstruktor tersebut adalah fungsi `onConstruct()` yang didalam fungsi tersebut akan mengecek apakah *session* pengguna masih aktif atau sudah tidak aktif. Apabila *session* pengguna masih aktif maka fungsi tersebut akan selalu mengecek hak akses pengguna apabila pengguna tersebut mengakses suatu modul. Hak akses tersebut merupakan `can_insert` (untuk melakukan penambahan data), `can_update` (untuk melakukan perubahan data), dan `can_delete` (untuk melakukan penghapusan data). Masing-masing hak akses tersebut memiliki suatu nilai yaitu 0 dan 1. Jika nilai pada hak akses tersebut bernilai 0 maka pengguna tidak memiliki hak akses dan jika 1 maka pengguna memiliki hak akses. Contoh kasus misalkan pengguna mengakses modul mengelola FRS dan memiliki nilai `can_insert` yaitu 1 maka pengguna tersebut dapat melakukan penambahan data pada modul mengelola FRS. Implementasi fungsi untuk mengecek

hak akses pengguna pada menu yang diakses adalah seperti pada Kode sumber 4.7.

```

1.  public function checkAccessControl()
2.  {
3.      $i = $this->session->get('i');
4.      $type = gettype($this->session->get('user')->roleunit);
5.      if ($type == 'array') {
6.          $role = $this->session->get('user')->roleunit[$i]->role_name;
7.      } else {
8.          $role = $this->session->get('user')->roleunit->role_name;
9.      }
10.
11.     $module = $this->router->getModuleName();
12.     $controller = $this->router->getControllerName();
13.     $path = './'.$module.'/'.$controller;
14.     $menus = $this->session->get('user')->menu;
15.
16.     if($module!='dashboard'){
17.         foreach($menus as $menu){
18.             if ($menu->path == $path && $role == $menu->role_name ) {
19.                 $this->can_insert = $menu->can_insert;
20.                 $this->can_update = $menu->can_update;
21.                 $this->can_delete = $menu->can_delete;
22.             }
23.         }
24.     }
25. }

```

Kode sumber 4.7 fungsi mengecek hak akses

4.3 Implementasi Routing

Untuk dapat mengakses SIAKAD pada *web browser* maka routing diperlukan. Routing yang diterapkan pada SIAKAD untuk menghubungkan modul, controller, dan action terkait. Format untuk routing adalah 'BaseURL/module/controller/{action}'. Jika action merupakan *indexAction* maka tidak perlu dituliskan, karena *indexAction* merupakan default action pada setiap controller. Routing diatur pada folder config, file routing.php. Modul oauth

merupakan modul yang default diatur pada SIAKAD, sehingga jika pengguna mengakses SIAKAD maka routing akan mengarahkan pengguna ke modul oauth, dalam hal ini yaitu halaman login utama SIAKAD. Routing akan merespon 404 not found jika route yang diminta tidak tersedia, Kode sumber 4.8 merupakan implementasi dari routing SIAKAD.

```

1.     foreach ($modules as $key => $module) {
2.         if ($key == "api") {
3.             include APP_PATH .'/modules/'.$key.'/config/routing.php';
4.         } else {
5.             $namespace = str_replace('Module','Controllers', $module["className"]);
6.             ;
7.             $router->add('/:key.:/:params', array(
8.                 'namespace' => $namespace,
9.                 'module' => $key,
10.                'controller' => 'index',
11.                'action' => 'index',
12.                'params' => 1
13.            ))->setName($key);
14.            $router-> add('/:key.:/:controller/:params', array(
15.                'namespace' => $namespace,
16.                'module' => $key,
17.                'controller' => 1,
18.                'action' => 'index',
19.                'params' => 2
20.            ));
21.            $router->add('/:key.:/:controller/:action/:params', array(
22.                'namespace' => $namespace,
23.                'module' => $key,
24.                'controller' => 1,
25.                'action' => 2,
26.                'params' => 3
27.            )); }

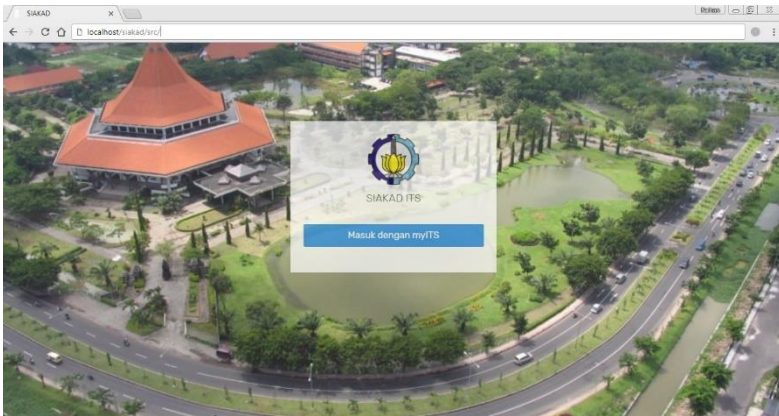
```

Kode sumber 4.8 implementasi routing

SIAKAD dapat menangani request router untuk setiap modul sebagai berikut.

1. BaseUrl/Module/Controller
2. BaseUrl/Module/Controller/Action

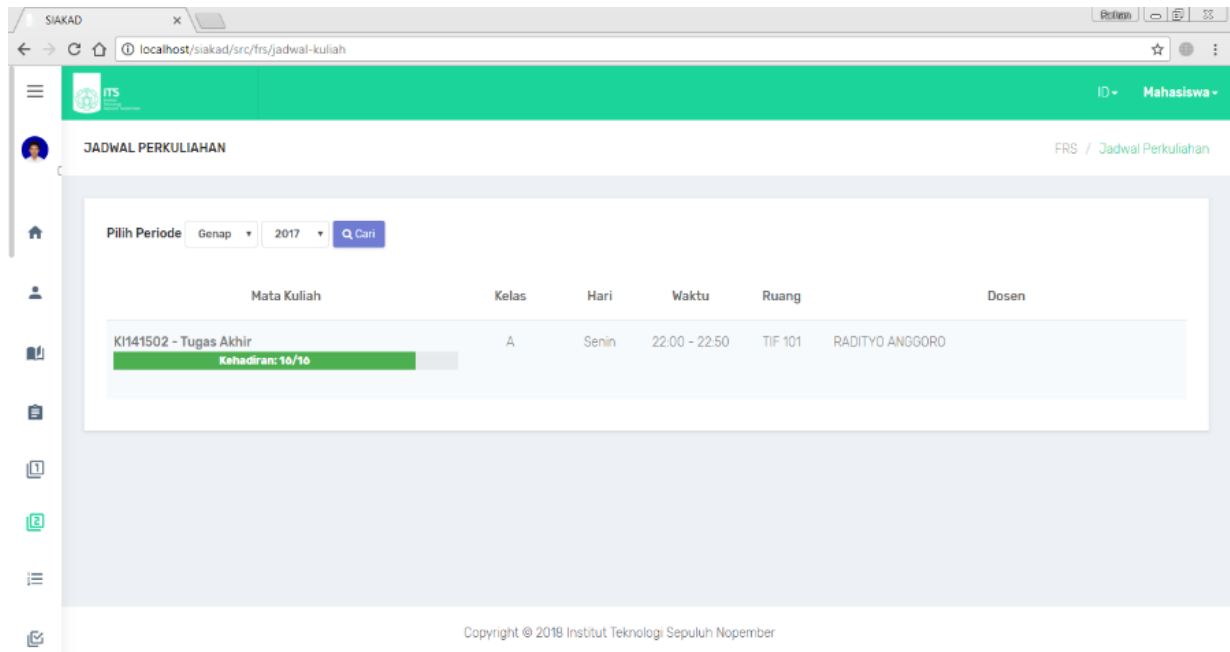
Gambar 4.2 merupakan contoh jika pengguna mengakses baseUrl Siakad. Pada contoh ini baseUrl adalah 'localhost/siakad/src'.



Gambar 4.2 halaman baseUrl

Berbeda dengan aturan routing untuk menangani request API, untuk dapat request API maka format yang ditentukan adalah 'BaseURL/api/versi API/Controller/action'. Sebagai contoh jika ingin mengakses API versi 1 dan untuk mendapatkan data daftar Gedung dari modul penjadwalan maka format yang digunakan adalah 'BaseUrl/api/v1/penjadwalan/daftargedung'.

Jika pengguna merupakan mahasiswa dan mengakses modul frs untuk melihat jadwal kuliah, maka routing yang diterapkan adalah 'baseUrl/frs/jadwal-kuliah'. Contoh seperti Gambar 4.3.



Gambar 4.3 halaman baseUrl/frs/jadwal-kuliah

Untuk mengimplementasikan *Repository Service Pattern*, maka setiap models pada modul ditempatkan disatu lokasi, pada SIAKAD models diletakkan pada folder config sehingga semua controller dapat mengakses models tersebut. Pada folder models terdapat folder service yang berguna untuk menangkap request dari controller dan mengolah data tersebut dengan logika yang diberikan untuk selanjutnya akan dikirimkan ke repository. *Query* untuk mengolah data ke basis data dilakukan pada repository. Untuk dapat menerapkan repository service pattern, semua controller yang berhubungan dengan basis data maka harus memanggil kelas service seperti pada Kode sumber 4.9.

```

1. public function indexAction()
2.     {
3.         $sar1 = Services::getService('Sar')->getAllSar1();
4.         $this->view->sar1=$sar1;
5.         $this->view->pick('sar1');
6.     }

```

Kode sumber 4.9 memanggil kelas service pada controller

Pada baris ke 3 dari Kode sumber 4.9, kelas indexAction memanggil kelas service Sar dan menunjuk ke fungsi getAllSar1(), *service* akan melanjutkan request dari *controller* tersebut ke *repository* seperti kode berikut.

```

1. public function getAllSar1()
2.     {
3.         return Repositories::getRepository('Sar')->getAllSar1();
4.     }

```

Kode sumber 4.10 memanggil kelas repository pada service

Selanjutnya kelas *repository* tersebut akan melakukan *query* untuk dapat berkomunikasi dengan basis data. kelas *repository* yang dipanggil dari kelas *service* sar fungsi getAllSar() adalah kelas

repository sar dan fungsi `getAllSar1()`. Adapun yang akan dilakukan oleh kelas *repository* sar fungsi `getAllSar()` adalah seperti pada Kode sumber 4.11.

```

1.  public function getAllSar1()
2.  {
3.      $sql ="SELECT sar1.*, s.nama as namasatker, jd.nama as nama
4.          from
5.          (SELECT DISTINCT id satker, id jenjang didik, sasaran nilai, rat
6.              aan_capaian
7.              from sar.sar INNER JOIN ref.[semester] smt
8.              on sar.id_smt = smt.id_smt
9.              where smt.a smt aktif= 1 and id level sar=1) sar1, satker s, ref.je
10.             njang_didik jd
11.             where sar1.id satker=s.id satker and sar1.id jenjang didik=jd.id j
12.             enjang_didik";
13.
14.      $sar1 = new Sar();
15.      return new Resultset(
16.          null,
17.          $sar1,
18.          $sar1->getReadConnection()->query($sql)
19.      ); }

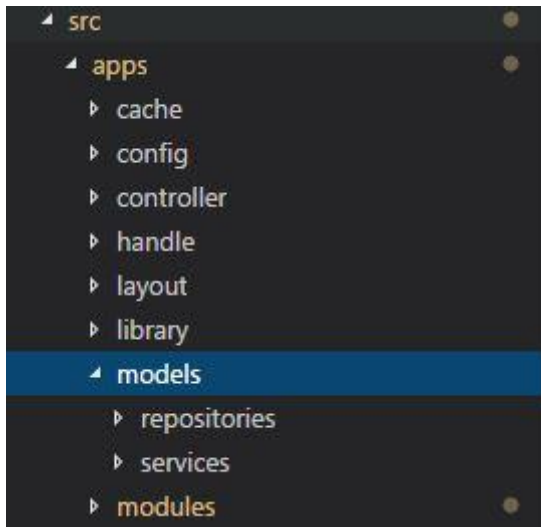
```

Kode sumber 4.11 raw query pada controller

Semua kelas *repository* akan menggunakan kelas `Phalcon\Mvc\Model` dan `extends` kelas tersebut.

4.4 Implementasi Repository Service Pattern

Penerapan *repository service pattern* dengan menaruh folder *models* tidak pada setiap modul, akan tetapi *models* ditempatkan pada folder apps, seperti pada Gambar 4.4.



Gambar 4.4 folder models

Terdapat dua folder di folder *models* pada Gambar 4.4 yaitu folder *service* dan folder *repository*. Pada folder *service* berisi semua file *service* yang berguna untuk mengolah logika yang berhubungan dengan basis data, sedangkan pada folder *repository* terdapat file-file *repository* yang berguna untuk berkomunikasi dengan basis data, pada *repository* file-file tersebut akan menggunakan kelas *models* dari Phalcon dan menggunakan *raw query*. Berikut merupakan contoh kode yang diterapkan pada service kelas Aktivitas.php seperti pada Kode sumber 4.12.

```

1. <?php
2.
3. namespace Siakad\Models\Services;
4.
5. use Siakad\Models\Repositories\Repositories;
6.
7. class Aktivitas

```

```

8. {
9.     public function getTingkatanKegiatan()
10.    {
11.        return Repositories::getRepository('Skem')->getTingkatanKegiatan();
12.    }
13. }

```

Kode sumber 4.12 file service pada kelas Aktivitas

Pada Kode sumber 4.12 fungsi `getTingkatanKegiatan()` memanggil kelas *repository* Skem dan fungsi `getTingkatanKegiatan()`. Kode sumber *repository* dari kelas Skem untuk fungsi `getTingkatanKegiatan()` adalah seperti berikut.

```

1.     <?php
2.
3.     namespace Siakad\Models\Repositories;
4.
5.     use Phalcon\Mvc\Model;
6.     use Phalcon\Mvc\Model\Resultset\Simple as Resultset;
7.
8.     class Aktivitas extends Model
9.     {
10.        public function getTingkatanKegiatan()
11.        {
12.            $db = $this->getDi()->getShared('db');
13.            $sql = $db->query(
14.                "SELECT id_tkt_keg, nama as tingkat FROM ref.[tingkat_kegi
                atan]");
15.            $data = $sql->fetchAll();
16.            return $data;
17.        } }

```

Kode sumber 4.13 repository pada kelas Skem

Kelas *repository* tersebut akan mengembalikan data ke kelas *service* untuk selanjutnya dikembalikan ke *controller* yang memanggil *service* tersebut. Untuk alur pemanggilan data pada *repository service pattern* ini sesuai pada Gambar 3.11.

4.5 Implementasi Modul Service Pada Kerangka Kerja

Pada Tugas Akhir ini, implementasi utama yang dilakukan adalah implementasi pada modul-modul service. Adapun implementasi yang telah dilakukan adalah OAuth2, RBAC, i18n, logging, API, dan *push notification*. Pada subbab ini akan menjelaskan implementasi yang telah dilakukan.

4.5.1 Implementasi OAuth2

Alur yang diimplementasikan untuk OAuth2 adalah seperti berikut.

1. Pengguna masuk ke halaman *login* SIAKAD.
2. Pengguna mengklik tombol *login* dengan myITS, maka SIAKAD akan mengirimkan parameter-parameter melalui URL ke myITS dengan *endpoint* /authorize.
3. myITS memeriksa parameter yang dikirimkan oleh SIAKAD.
 - a. apabila parameter yang dikirimkan SIAKAD sesuai dengan parameter yang telah didaftarkan pada myITS maka lanjut ke tahap selanjutnya.
 - b. Apabila parameter yang dikirimkan SIAKAD tidak sesuai dengan parameter yang telah didaftarkan pada SIAKAD, maka myITS akan mengirimkan pesan *error*.
4. myITS menampilkan halaman *login*.
5. Pengguna melakukan tahap otentikasi dengan memasukkan *username* dan *password* pada myITS.
6. myITS memeriksa apakah *username* dan *password* valid atau tidak.
 - a. Jika valid, maka akan dilanjutkan pada tahap selanjutnya.
 - b. Jika tidak valid, maka kembali ke tahap 4.
7. Sistem meminta akses token ke myITS melalui *endpoint* /token.

8. myITS memberikan akses token ke SIAKAD.
9. SIAKAD akan melakukan validasi token dengan menggunakan *public key* yang didapatkan pada saat mendaftarkan SIAKAD di myITS.
 - a. Jika token valid, maka akan dilanjutkan ke tahap selanjutnya.
 - b. Jika token tidak valid, maka pengguna akan diarahkan kembali ke halaman *login* myITS, kembali ke tahap 4.
10. SIAKAD akan meminta informasi data-data pengguna ke myITS melalui *endpoint* /userinfo dengan menggunakan token yang telah divalidasi.
11. myITS memberikan informasi pengguna sesuai dengan *scope* yang diminta oleh SIAKAD.
12. SIAKAD akan menyimpan semua informasi pengguna ke *session*.

Untuk dapat masuk ke SIAKAD, pengguna akan diarahkan ke halaman myITS. Pada saat pengguna mengklik tombol login pada halaman login SIAKAD maka sistem akan mengirimkan informasi ke myITS melalui url seperti berikut.

'http://my.its.id/authorize?response_type=code&scope=openid%20profile%20email%20phone%20roleunit%20menu%20resource&client_id=A7600479-013E-4377-A320-602281F8D286&state=khz&prompt=login&redirect_uri=http://localhost/siakad/src/oauth/Authorize'.

Adapun detail dari informasi pada url tersebut adalah seperti pada Tabel 4.4.

Tabel 4.4 Parameter Url pada endpoint /authorize

Parameter	Value
Response_type	Code
Scope	Openid, profile, email, phone, roleunit, menu, resource
Client_id	A7600479-013E-4377-A320-602281F8D286
State	khz
prompt	login
redirect_uri	http://localhost/siakad/src/oauth/Authorize

Penjelasan dari setiap parameter yang dikirimkan SIAKAD ke myITS adalah seperti Tabel 4.5.

Tabel 4.5 Deskripsi parameter pada Url endpoint /authorize

Parameter	Deskripsi
Response_type	Pada SIAKAD <i>value</i> yang dikirimkan adalah code, karena aplikasi SIAKAD menggunakan <i>authorization code flow</i>
Scope	Parameter scope dapat diisi lebih dari satu nilai. Nilai wajib yang harus diisi adalah openid, dimana openid akan mengembaikan nilai sub (id user) ke SIAKAD. Nilai lainnya yang dikirimkan oleh SIAKAD adalah seperti berikut.

Parameter	Deskripsi
	<ol style="list-style-type: none"> 1. Profile, nilai ini akan mengembalikan informasi mengenai nilai <i>name</i>, <i>nickname</i>, <i>username</i>, <i>picture</i>, <i>gender</i>, <i>birthdate</i>, <i>zoneinfo</i>, <i>locale</i>, <i>last_update</i>, dan <i>integra_id</i> pengguna. 2. Email, nilai ini akan mengembalikan informasi mengenai nilai <i>email</i>, <i>email_verified</i>, <i>alternate_email</i>, dan <i>alternate_email_verified</i> pengguna. 3. Phone, nilai ini akan mengembalikan informasi mengenai nilai <i>phone</i>, dan <i>phone_verified</i> pengguna. 4. Roleunit, nilai ini akan mengembalikan informasi mengenai nilai <i>role_id</i>, <i>role_name</i>, <i>unit_id</i>, <i>unit_name</i>, dan <i>role_default</i> pengguna. 5. Menu, nilai ini akan mengembalikan informasi mengenai nilai <i>menu_id</i>, <i>parent_id</i>, <i>menu_name</i>, <i>name_en</i>, <i>path</i>, <i>can_insert</i>, <i>can_update</i>, <i>can_delete</i>, <i>menu_order</i>, <i>role_name</i>, dan <i>icon</i> menu yang dapat diakses pengguna.

Parameter	Deskripsi
Client_id	Merupakan id unik yang dimiliki oleh setiap client. Nilai ini akan didapatkan oleh klien pada saat mendaftarkan aplikasi pada aplikasi myITS.
State	Nilai pada parameter state merupakan nilai yang berisi random string. Pada SIAKAD string yang dikirimkan pada parameter state terdiri dari 3 karakter. Nilai state ini akan digunakan oleh SIAKAD untuk memeriksa apakah authorization response yang diterima valid atau tidak.
prompt	<p>Nilai pada parameter prompt yaitu.</p> <ol style="list-style-type: none"> 1. Login, menandakan bahwa pengguna harus melakukan otentikasi dengan memasukkan username dan password. Sistem akan menampilkan halaman login untuk pengguna. Nilai ini merupakan nilai default jika parameter tidak disertakan pada <i>authorization request</i>. 2. None, menandakan bahwa pengguna tidak perlu melakukan proses otentikasi. Saat menerima authorization request dengan nilai prompt=none, maka sistem akan memeriksa session pengguna. Jika valid, maka akan langsung menampilkan halaman dashboard

Parameter	Deskripsi
	myITS dan jika tidak maka pesan error “login required” akan dikembalikan ke <i>redirect_uri</i> klien.
redirect_uri	Nilai dari parameter ini adalah <i>redirect_uri</i> yang disimpan dalam sistem. Redirect_uri ini akan digunakan untuk aplikasi klien (SIKAD) menangani response.

Pada saat parameter diatas dikirimkan ke myITS, maka myITS akan memeriksa apakah parameter yang dikirimkan *valid* atau tidak, jika tidak valid maka pengguna tidak dapat masuk menggunakan myITS dan jika valid maka pengguna akan diarahkan ke halaman *login* myITS. Kode sumber 4.14 merupakan contoh pada saat SIKAD meminta akses token ke myITS melalui *endpoint /token*.

```

1. public function sendTokenRequestAction($code)
2. {
3.     $getcontent = $this->getContentFileAction()->requestContent;
4.     $getcontent['code'] = $code;
5.     $content = http_build_query($getcontent);
6.     $result = Services::getService('Oauth')-
       >sendTokenRequestAction($content);
7.     return $result;
8. }
```

Kode sumber 4.14 fungsi meminta akses token

Pada fungsi tersebut, variabel *\$getcontent* merupakan variabel yang mengambil informasi yang terdapat pada file *config.php*. informasi yang diperlukan adalah sebagai berikut.


```

1. 'requestContent' => [
2.     'client_id' => 'A7600479-013E-4377-A320-602281F8D286' ,
3.     'client_secret' => 'f131c93eeb600f4a2d8b2e5b329245fd',
4.     'grant_type' => 'authorization_code',
5.     'redirect_uri' => 'http://localhost/siakad/src/oauth/Authorize'
6. ]

```

Kode sumber 4.15 parameter Url pada config.php

Selanjutnya, fungsi tersebut akan mengirimkan data-data tersebut ke endpoint /token melalui service OAuth. Service OAuth yang diimplementasikan adalah seperti pada Kode sumber 4.16.

```

1. public function sendTokenRequestAction($content)
2. {
3.     $ch = curl_init();
4.     curl_setopt($ch, CURLOPT_URL, "http://my.its.id/token");
5.     curl_setopt($ch, CURLOPT_POST, 1);
6.     curl_setopt($ch, CURLOPT_POSTFIELDS, $content);
7.     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
8.     $result = curl_exec ($ch);
9.     curl_close ($ch);
10.
11.     return $result = json_decode($result);
12. }

```

Kode sumber 4.16 service untuk mengirimkan parameter ke endpoint /token

Data-data tersebut akan dikirimkan dalam format JSON melalui curl dengan metode POST. Selanjutnya, myITS akan mengirimkan akses token ke SIAKAD yang selanjutnya digunakan untuk meminta informasi pengguna. Permintaan informasi pengguna ke myITS melalui endpoint /userinfo. Kode sumber 4.17 merupakan implementasi fungsi untuk mendapatkan informasi pengguna.

```

1. public function getUserInfoAction($result)
2. {
3.     $userInfo = Services::getService('OAuth')->getUserInfoAction($result);

```

```

4.     return $userInfo;
5. }

```

Kode sumber 4.17 fungsi meminta informasi pengguna

Pada fungsi tersebut, variabel \$userInfo memanggil kelas service Oauth dengan fungsi getUserInfoAction(\$result) dan mengirimkan parameter \$result yang berupa token yang didapatkan pada tahap sebelumnya. Kelas service Oauth pada fungsi getUserInfoAction(\$result) diimplementasikan seperti pada Kode sumber 4.18.

```

1. public function getUserInfoAction($result)
2. {
3.     $ch2 = curl_init();
4.     curl_setopt($ch2, CURLOPT_URL, "http://my.its.id/userinfo");
5.     curl_setopt($ch2, CURLOPT_HTTPHEADER, array(
6.         'Authorization: Bearer '.$result->access_token
7.     ));
8.     curl_setopt($ch2, CURLOPT_RETURNTRANSFER, true);
9.     $output = curl_exec ($ch2);
10.    curl_close ($ch2);
11.    $output = strstr($output, '{');
12.    $output = json_decode($output);
13.    return $output;
14. }

```

Kode sumber 4.18 service meminta informasi pengguna pada endpoint /userinfo

Setelah tahap tersebut dilakukan, maka myITS akan mengirimkan informasi pengguna dan selanjutnya SIAKAD akan menyimpan informasi pengguna tersebut ke dalam session. Kode sumber 4.19 merupakan implementasi untuk menyimpan informasi pengguna ke session.

```

1. public function setSessionAction($result)
2. {
3.     $this->session->set('user', $this->userInfo);
4.     $this->session->set('payload', $this->payload);
5.     $this->session->set('token', $result->access_token);
6. }

```

Kode sumber 4.19 menyimpan informasi pengguna ke session

Selanjutnya SIAKAD akan mengarahkan pengguna ke halaman dashboard pengguna sesuai dengan hak akses yang ada pada pengguna. Jika hak akses hanya 1 maka pengguna langsung diarahkan ke dashboard hak akses tersebut, dan jika pengguna memiliki hak akses lebih dari 1, maka SIAKAD akan mengarahkan pengguna ke dashboard dengan hak akses yang pertama yang terdapat pada data hak akses pada myITS. Kode sumber 4.20 merupakan implementasi untuk mengarahkan pengguna ke halaman dashboard SIAKAD.

```

1. public function redirectDashboardAction()
2. {
3.     if ($this->request->isPost()) {
4.         $index = $_POST['role'];
5.         $this->session->set('i', $index);
6.         $this->changeSessionAction();
7.     } else {
8.         $this->session->set('i', 1);
9.         $i = $this->session->get('i');
10.        $type = gettype($this->userInfo->roleunit);
11.        if ($type == 'array') {
12.            $role = str_replace(" ", "-", strtolower($this->session->get('user')-
13.                >roleunit[$i->role_name]));
14.        } else {
15.            $role = str_replace(" ", "-", strtolower($this->session->get('user')-
16.                >roleunit->role_name));
17.        }
18.    }
19. }

```

```

16. $this->logger->login("Masuk ke aplikasi SIAKAD sebagai ".$role);
17.     $this->response->redirect("dashboard/".$role);
18. } }

```

Kode sumber 4.20 fungsi mengarahkan pengguna ke dashboard

4.5.2 Implementasi RBAC

Untuk dapat mengatur hak akses pada SIAKAD dan mendefinisikan semua menu-menu yang ada pada SIAKAD, maka *developer* harus mendefinisikan hal tersebut pada myITS. Gambar 4.5 merupakan halaman untuk mengatur hak akses pengguna dan Gambar 4.6 merupakan halaman untuk menambahkan menu.

Setelah pengguna berhasil melakukan tahap otentikasi pada myITS, salah satu informasi yang didapatkan pada endpoint `/userinfo` adalah menu dan hak akses pengguna. Hak akses pengguna memiliki nilai diantara lain adalah dosen, mahasiswa, kepala prodi, dll. Sedangkan menu merupakan informasi mengenai menu-menu yang dapat diakses oleh pengguna tersebut berdasarkan hak akses pengguna. Kode sumber 4.21 merupakan implementasi untuk mengarahkan pengguna ke halaman sesuai dengan hak akses pengguna yaitu `role_name` pada informasi pengguna yang didapatkan dari myITS.

Jika role dosen maka pengguna akan diarahkan ke dashboard dosen, jika role mahasiswa maka pengguna akan diarahkan ke dashboard mahasiswa, begitu juga seterusnya. Setelah pengguna diarahkan ke halaman dashboard sesuai dengan hak aksesnya maka pengguna juga akan mendapatkan menu-menu yang dapat diakses pada sidebar halaman dashboard pengguna. Gambar 4.7 merupakan implementasi untuk sidebar untuk hak akses pengguna dosen dan Gambar 4.8 hak akses pengguna mahasiswa.

Tambah User

Nama*wajib

Nama user

Nickname

Nama panggilan user

Username*wajib

Username

Email*wajib

Email user

Alternate Email

Email alternatif user

Telepon

62

ex: 6281652716251

Tanggal Lahir

mm/dd/yyyy

Password*wajib

Min. 6 karakter

Ulangi Password*wajib

Ulangi password

Gender *wajib

Pilih gender

Status Keaktifan

Aktif

Locale

ID

Zoneinfo

Asia/Jakarta

Tipe User*wajib

Mahasiswa

Sandbox

False

Foto Profil

Drag and drop a file here or click

Batal

Submit

Gambar 4.5 antar muka halaman menambahkan data user

Tambah Menu ✕

Nama Menu*wajib

Nama Menu (Bhs. Inggris)

Parent Menu

Client

Deskripsi

Path*wajib

Icon

Order

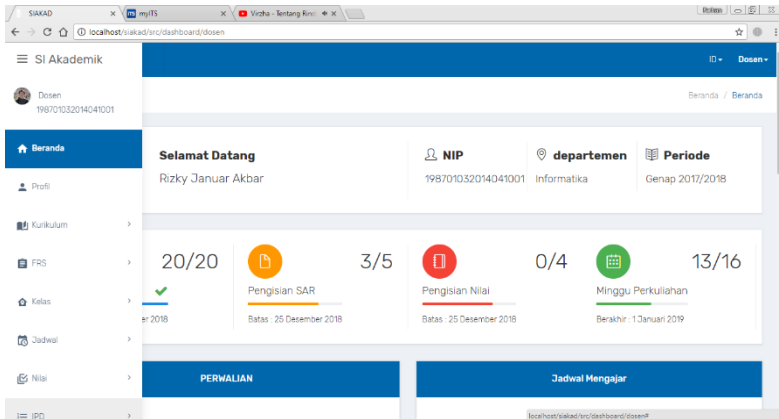
Visibility

Status

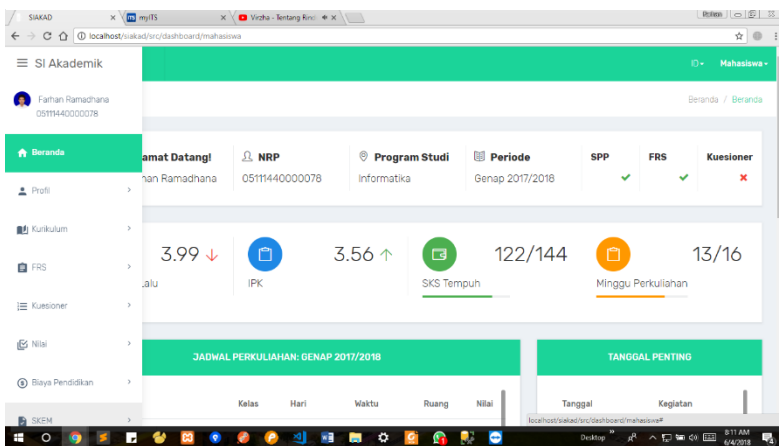
Gambar 4.6 antar muka halaman menambahkan menu pada myITS

1. `$this->response->redirect("dashboard/".$role);`

Kode sumber 4.21 Implementasi untuk mengarahkan pengguna ke dashboard sesuai hak akses



Gambar 4.7 dashboard pengguna dosen



Gambar 4.8 dashboard pengguna mahasiswa

Kode sumber 4.22 merupakan contoh informasi pengguna yang diperoleh dari myITS. informasi pada /userinfo pada penulisan buku ini dipecah menjadi dua bagian, yaitu bagian yang pertama merupakan bagian mengenai informasi pribadi pengguna dan

bagian kedua merupakan contoh menu yang dapat diakses oleh pengguna.

```

1.  {
2.    "email": "rizky@if.its.ac.id",
3.    "email_verified": "0",
4.    "alternate_email": "ricesky@gmail.com",
5.    "alternate_email_verified": "0",
6.    "phone": "6281234567890",
7.    "phone_verified": "1",
8.    "name": "Rizky Januar Akbar",
9.    "nickname": "Rizky",
10.   "username": "rizkyja",
11.   "gender": "L",
12.   "birthdate": "1970-01-01",
13.   "zoneinfo": "Asia/Jakarta",
14.   "locale": "ID",
15.   "last_update": "2018-05-21 12:26:12.000",
16.   "integra_id": null,
17.   "resource": {
18.     "resource_id": null,
19.     "name": null,
20.     "path": null,
21.     "method": null
22.   },
23.   "roleunit": [
24.     {
25.       "role_id": "6",
26.       "role_name": "Dosen",
27.       "unit_id": "3",
28.       "unit_name": "S1 Informatika",
29.       "role_default": "0"
30.     },
31.     {
32.       "role_id": "12",
33.       "role_name": "Kepala Prodi",
34.       "unit_id": "3",
35.       "unit_name": "S1 Informatika",
36.       "role_default": "0"

```



```

37.     }
38.   ],
39.   "sub": "A1E764DC-205C-4DC6-B6F0-C6C0D8762318"

```

Kode sumber 4.22 informasi pengguna pada API myITS

Pada informasi pengguna, didapatkan informasi diantaranya mengenai nama, email, *locale*, *roleunit*, dll. Pada contoh Kode sumber 4.2, pengguna tersebut memiliki dua hak akses yaitu hak akses sebagai dosen dan hak akses sebagai kepala prodi dan pengguna memiliki nilai “ID” pada parameter *locale* yang menandakan bahasa pengantar *default* yang akan ditampilkan SIAKAD adalah bahasa Indonesia. Kode sumber 4.23 merupakan menu-menu yang dapat diakses oleh pengguna yang merupakan dosen.

```

1.   "menu": [
2.     {
3.       "menu_id": "17",
4.       "parent_id": null,
5.       "menu_name": "Beranda",
6.       "name_en": "Home",
7.       "path": "../dashboard/dosen",
8.       "can_insert": "0",
9.       "can_update": "0",
10.      "can_delete": "0",
11.      "menu_order": "1",
12.      "role_name": "Dosen",
13.      "icon": "mdi mdi-home fa-fw"
14.    },
15.    {
16.      "menu_id": "20",
17.      "parent_id": "19",
18.      "menu_name": "Data Mata Kuliah",
19.      "name_en": "List of Courses",
20.      "path": "../kurikulum/data-mk",
21.      "can_insert": "0",
22.      "can_update": "0",
23.      "can_delete": "0",

```

```

24.     "menu_order": "1",
25.     "role_name": "Dosen",
26.     "icon": "mdi mdi-numeric-1-box-multiple-outline fa-fw"
27.   },
28.   {
29.     "menu_id": "23",
30.     "parent_id": "22",
31.     "menu_name": "Perwalian",
32.     "name_en": "Guardian",
33.     "path": "../frs/perwalian-dosen",
34.     "can_insert": "0",
35.     "can_update": "0",
36.     "can_delete": "0",
37.     "menu_order": "1",
38.     "role_name": "Dosen",
39.     "icon": "mdi mdi-numeric-1-box-multiple-outline fa-fw"
40.   },

```

Kode sumber 4.23 informasi menu pengguna pada myITS

Semua informasi pengguna tersebut disimpan ke session dan menu-menu tersebut akan diimplementasikan sesuai dengan informasi yang diberikan.

4.5.3 Implementasi Internationaliation (i18n)

Salah satu fitur yang disediakan pada SIAKAD yaitu internationalization (i18n) ataupun penyediaan bahasa pengantar yang terdiri dari bahasa Indonesia dan bahasa Inggris. Bahasa pengantar yang ditampilkan pada saat pengguna masuk ke dashboard SIAKAD adalah bahasa yang didapatkan pada informasi pengguna yang dikirimkan oleh aplikasi myITS. Aplikasi myITS akan mengirimkan informasi pengguna salah satunya *locale*, dimana *locale* tersebut memberikan informasi bahasa default yang digunakan oleh pengguna. Jika *locale* berisi nilai “ID” maka pengguna akan ditampilkan bahasa pengantar dalam bahasa Indonesia dan jika *locale* berisi nilai “EN” maka pengguna akan ditampilkan bahasa pengantar dalam bahasa Inggris. Penentuan bahasa tersebut dilakukan pada saat pengguna

masuk ke aplikasi SIAKAD, pengguna juga dapat mengubah bahasa pengantar pada SIAKAD, dimana opsi tersebut terdapat pada *topbar* dashboard pengguna. Berikut implementasi yang dilakukan pada i18n pada saat myITS mengirimkan informasi pengguna ke SIAKAD.

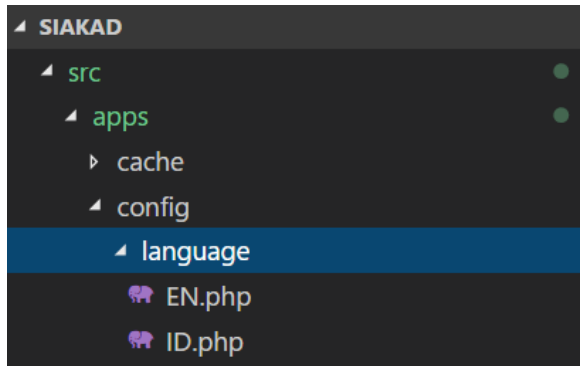
1. SIAKAD menyimpan informasi pengguna salah satunya yaitu berupa *locale* ke session.
2. SIAKAD akan mengarahkan ke controller sesuai dengan hak akses pengguna
3. Controller yang menggunakan kelas Phalcon\Translate\Adapter\NativeArray akan memanggil fungsi `getLanguageAction()`.
4. Fungsi `getLanguageAction()` akan membaca session yang berisi informasi pengguna yaitu *locale*
5. Fungsi `getLanguageAction()` membaca file ID atau EN sesuai dengan informasi pada *locale* dan mengembalikan nilai ke controller

Kode sumber 4.24 merupakan implementasi fungsi untuk menyimpan session bahasa yang digunakan pada `AuthorizeController`.

```
1. public function setLanguageAction($lang,$link)
2. {
3.     $link = str_replace("_","/", $link);
4.     $this->session->set('language',$lang);
5.     $this->response->redirect('".$link);
6. }
```

Kode sumber 4.24 fungsi menyimpan informasi bahasa pengantar

File yang berisikan bahasa bahasa yang terdapat pada SIAKAD terdapat pada folder `config/language`. Seperti pada Gambar 4.9.



Gambar 4.9 folder language pada SIAKAD

Kode sumber 4.25 merupakan implementasi pada file bahasa Indonesia yang terdapat pada SIAKAD.

```

1. <?php
2.
3. messages = [
4. 'selamatDatang' => 'Selamat Datang',
5. 'NIP' => 'NIP',
6. 'Departemen' => 'Departemen',
7. 'periode' => 'Periode',
8. 'batas' => 'Batas',
9. 'mingguPerkuliahan' => 'Minggu Perkuliahan',
10. 'berakhir' => 'Berakhir',
11. 'pengumuman' => 'Pengumuman',
12. 'genap' => 'Genap',
13. 'ganjil' => 'Ganjil',
14. 'jadwalMengajar' => 'Jadwal Mengajar',
15. 'nama' => 'Nama',
16. 'nrp' => 'NRP',
17. 'ips' => 'IPS',
18. 'ipk' => 'IPK',
19. 'sksDiambil' => 'SKS Diambil',
20. 'perwalian' => 'Perwalian',
21. 'disetujui' => 'Disetujui',
22. 'belumDisetujui' => 'Belum Disetujui',

```

```

23. 'senin' => 'Senin',
24. 'selasa' => 'Selasa',
25. 'rabu' => 'Rabu',
26. 'kamis' => 'Kamis',
27. 'jumat' => 'Jumat',
28. 'sabtu' => 'Sabtu',
29. 'mataKuliah' => 'Mata Kuliah',
30. 'kelas' => 'Kelas',
31. 'waktu' => 'Waktu',
32. 'ruang' => 'Ruang',
33. ];

```

Kode sumber 4.25 file terjemahan bahasa Indonesia

Kode sumber 4.26 merupakan implementasi pada file bahasa Inggris yang terdapat pada SIAKAD.

```

1. <?php
2.
3. $messages = [
4.     'selamatDatang' => 'Welcome',
5.     'nip' => 'ID',
6.     'departemen' => 'Department',
7.     'periode' => 'Period',
8.     'batas' => 'Deadline',
9.     'mingguPerkuliahan' => 'Week Of Study',
10.    'berakhir' => 'End',
11.    'pengumuman' => 'Announcement',
12.    'genap' => 'Even',
13.    'ganjil' => 'Odd',
14.    'jadwalMengajar' => 'Lecturing Schedule',
15.    'nama' => 'Name',
16.    'nrp' => 'NRP',
17.    'ips' => 'GP',
18.    'ipk' => 'GPA',
19.    'sksDiambil' => 'Taken Credits',
20.    'perwalian' => 'Approval',
21.    'disetujui' => 'Approved',
22.    'belumDisetujui' => 'Waiting For Approval',
23.    'senin' => 'Monday',
24.    'selasa' => 'Tuesday',
25.    'rabu' => 'Wednesday',

```

```

26. 'Kamis' => 'Thursday',
27. 'Jumat' => 'Friday',
28. 'Sabtu' => 'Saturday',
29. 'mataKuliah' => 'Subject',
30. 'kelas' => 'Class',
31. 'waktu' => 'Time',
32. 'ruang' => 'Room',
33. ];

```

Kode sumber 4.26 file terjemahan bahasa Inggris

Pada variabel \$messages, data-data tersebut berupa array dimana parameter disebelah kiri menyatakan variabel yang nantinya akan dipakai pada view dan sebelah kanan merupakan parameter nilai dari variabel tersebut. Pada setiap halaman di SIAKAD, untuk setiap elemen yang dapat berubah-ubah bahasanya maka elemen tersebut harus diimplementasikan dengan variabel, bukan dengan *hardcode*. Kode sumber 4.27 merupakan contoh implementasi untuk mencetak kata selamat datang pada SIAKAD.

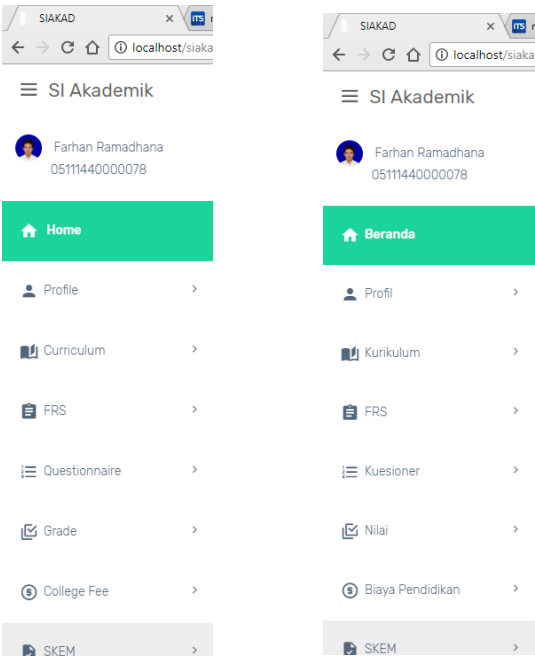
```

1. <h3 style="font-weight: bold;">
2.     <?php echo $t->_('selamatDatang'); ?>
3. </h3>

```

Kode sumber 4.27 mencetak variabel bahasa ke tampilan

Nilai pada variabel \$t->_('selamatDatang') tersebut dapat berganti sesuai dengan pilihan bahasa yang ditentukan oleh pengguna. Gambar 4.10 merupakan implementasi halaman pengguna menggunakan bahasa Indonesia dan bahasa Inggris.



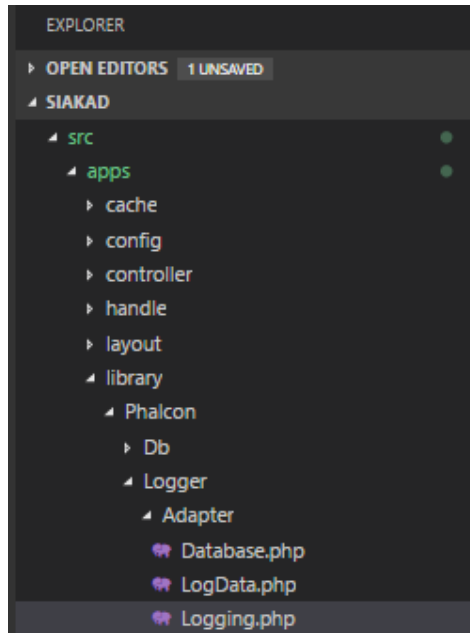
(a)

(b)

Gambar 4.10 sidebar (a) bahasa Inggris; (b) bahasa indonesia

4.5.4 Implementasi Logging

Untuk dapat mencatat setiap aktivitas yang dilakukan oleh pengguna pada SIAKAD, maka SIAKAD menerapkan *logging*. logging diimplementasikan dengan menggunakan kelas Phalcon\Logger\AdapterInterface. File untuk logging terdapat pada folder library seperti pada Gambar 4.11.



**Gambar 4.11 folder library
logger**

Setiap aktivitas yang ingin dicatat akan dicatat pada setiap controller setiap modul. Apabila suatu controller ingin mencatat aktivitas yang dilakukan oleh pengguna maka kelas controller tersebut harus mengimplementasikan kelas Phalcon\Logger\Adapter\Logging. Implementasi yang dilakukan untuk mencatat aktivitas log adalah seperti Kode sumber 4.28.


```
1. $this->logger->login("Masuk ke aplikasi SIAKAD sebagai ".$role);
```

Kode sumber 4.28 mencatat log aktivitas

Kelas log terdapat banyak kategori untuk mencatat aktivitas log. Pada contoh diatas kategori yang digunakan adalah *login*. Pada SIAKAD setiap kategori tersebut diberikan sebuah id kategori dimana id kategori tersebut merupakan sebuah bilangan untuk membedakan antara suatu kategori dengan kategori lainnya. Kode sumber 4.29 merupakan kategori-kategori log yang dapat diimplementasikan pada SIAKAD.

```
1. public function login($message)
2. {
3.     $category = 1;
4.     $this->insertLog($message,$category);
5. }
6.
7. public function debug($message, array $context = NULL)
8. {
9.     $category = 9;
10.    $this->insertLog($message,$category);
11. }
12.
13. public function error($message, array $context = NULL)
14. {
15.     $category = 10;
16.     $this->insertLog($message,$category);
17. }
18.
19. public function info($message, array $context = NULL)
20. {
21.     $category = 11;
22.     $this->insertLog($message,$category);
23. }
24.
25. public function notice($message, array $context = NULL)
26. {
27.     $category = 12;
28.     $this->insertLog($message,$category);
```

```

29. }
30.
31. public function warning($message, array $context = NULL)
32. {
33.     $category = 13;
34.     $this->insertLog($message,$category);
35. }
36.
37. public function alert($message, array $context = NULL)
38. {
39.     $category = 14;
40.     $this->insertLog($message,$category);
41. }
42.
43. public function emergency($message, array $context = NULL)
44. {
45.     $category = 15;
46.     $this->insertLog($message,$category);
47. }

```

Kode sumber 4.29 fungsi untuk insert log

Aktivitas yang dicatat pada setiap *controller* akan dilanjutkan ke kelas logging dan selanjutnya akan dilanjutkan ke kelas LogData untuk dicatat semua informasi yang akan dicatat, Kode sumber 4.30 merupakan data-data yang akan dicatat oleh LogData untuk selanjutnya dikirimkan ke kelas Log service.

```

1. $data = array(
2.     "client_id" => $client_id,
3.     "user_id" => $user_id,
4.     "category" => $category,
5.     "activity" => $activity,
6.     "time" => $time,
7.     "ip_address" => $ip_address,
8.     "hostname" => $hostname,
9.     "os" => $os,
10.    "browser" => $this->getBrowser()
11. );

```

```
12. $data = json_encode($data);
```

Kode sumber 4.30 data log

Data pada LogData akan di bungkus dalam bentuk JSON dan data tersebut dikirimkan ke service. Kode sumber 4.31 merupakan implementasi untuk kelas service Log.

```
1. public function insertLogAction($data, $token)
2. {
3.     $headers = array(
4.         'Authorization: Bearer '.$token,
5.         'Content-Type: application/json'
6.     );
7.
8.     $ch = curl_init();
9.     curl_setopt($ch, CURLOPT_URL, "http://my.its.id/api/v1/log");
10.    curl_setopt($ch, CURLOPT_POST, true);
11.    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
12.    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
13.    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
14.
15.    $output = curl_exec ($ch);
16.
17.    curl_close ($ch);
18.
19.    return $output;
20. }
```

Kode sumber 4.31 fungsi untuk mengirimkan log

Pada kelas *service* logging diatas, data akan dikirimkan ke basis data myITS untuk disimpan. Data akan dikirim melalui API yang telah disediakan oleh myITS melalui endpoint /api/v1/log.

4.5.5 Implementasi API

API digunakan untuk pertukaran data antara SIAKAD dengan aplikasi lainnya. Api dapat diimplementasikan ke dalam beberapa versi, pada saat ini versi 1 pada API sudah dapat diakses, untuk versi 2 dan seterusnya tergantung kebutuhan kedepannya. Untuk dapat mendaftar versi API maka *developer* akan mendefinisikan versi API tersebut ke loader.php dan modules.php. Kode sumber 4.32 merupakan implementasi untuk mendefinisikan API pada loader.php

```
1. 'Siakad\Api' => APP_PATH . '/modules/api/v1',
2. 'Siakad\Api2' => APP_PATH . '/modules/api/v2',
```

Kode sumber 4.32 versi API

Kode sumber 4.33 merupakan implementasi mendefinisikan API pada modules.php

```
1. 'api/v1' => [
2.     'className' => 'Siakad\Api\Module',
3.     'path'      => APP_PATH . '/modules/api/v1/Module.php'
4. ],
5. 'api/v2' => [
6.     'className' => 'Siakad\Api2\Module',
7.     'path'      => APP_PATH . '/modules/api/v2/Module.php'
8. ],
```

Kode sumber 4.33 mendaftarkan modul API

Setelah modul-modul tersebut didaftar pada loader.php dan modules.php maka *developer* harus mendefinisikan Url untuk dapat mengakses API tersebut. Kode sumber 4.34 merupakan implementasi untuk mendefinisikan Url dari API.

```
1. <?php
2.
3. $versions = [
```

```

4.     'v1' => '/api/v1',
5.     'v2' => '/api/v2'
6. ];
7.
8. $users = new \Phalcon\MvcRouter\Group(array(
9.     'module' => 'api',
10.    //'controller' => 'users'
11. ));
12. $users->setPrefix($versions['v1'].'/users/:controller/:action');
13. $users->addGet("", array(
14.     'module' => 'api/v1',
15.     'controller' => 2,
16.     'action' => 2
17. ));
18. $users->setPrefix($versions['v2'].'/users/:controller/:action');
19. $users->addGet("", array(
20.     'module' => 'api/v2',
21.     'controller' => 1,
22.     'action' => 2
23. ));
24. $router->mount($users);

```

Kode sumber 4.34 routing modul API

Ketika pengguna mengakses API, maka pengguna diharuskan mengirimkan data header yang berupa id token berupa JWT pada parameter “X-jwt-assertion”. SIAKAD akan memeriksa token yang dikirimkan apakah valid atau tidak. Jika token valid dan endpoint yang diakses benar maka SIAKAD akan memberikan response sesuai dengan permintaan pengguna, jika tidak maka SIAKAD akan menampilkan pesan 401 (unauthorized). Kode sumber 4.35 merupakan fungsi untuk menangkap request jika pengguna mengakses endpoint /api/v1/users/

```

1. public function indexAction()
2. {
3.     $validate = $this->validate();
4.     $checkInfo = $this->getUserID();
5.     if ($_SERVER['REQUEST_METHOD'] == 'GET') {

```

```

6.         if ($validate == 'validate_success') {
7.             $get = Services::Service(Users)->getAllUsers();
8.             $data = [];
9.             $this->response->setJsonContent($get);
10.            $this->response->send();
11.        }
12.    } else{
13.        $this->throwError(400,'Bad request');
14.    }
15. }

```

Kode sumber 4.35 contoh endpoint API

Kode sumber 4.36 merupakan implementasi fungsi untuk melakukan validasi terhadap request dan parameter yang dikirimkan oleh pengguna.

```

1. public function validate()
2. {
3.     $this->token = $this->getAuthorizationHeader();
4.     $this->pubkey = $this->getPublicKey();
5.
6.     try {
7.         $this->payload = JWT::decode($this->token, $this-
8.         >pubkey, array('RS256'));
9.         return $status = 'validate_success';
10.    } catch (\Exception $e) {
11.        $this->throwError(401,'Invalid Token ('.$e->getMessage().')');
12.    }

```

Kode sumber 4.36 fungsi validasi id token

Fungsi tersebut akan memanggil fungsi `getAuthorizationHeader()` yang berfungsi untuk mengambil nilai id token yang dikirimkan pengguna melalui parameter `X-jwt-assertion`. Kode sumber 4.37 merupakan implementasi pada fungsi `getAuthorizationHeader()`.

```

1. public function getAuthorizationHeader()
2.     {
3.         $headers = null;
4.         if (function_exists('apache_request_headers')) {
5.             $requestHeaders = apache_request_headers();
6.             $requestHeaders = array_combine(array_map('ucwords', array_keys
($requestHeaders)), array_values($requestHeaders));
7.
8.             if (isset($requestHeaders["X-jwt-assertion"])) {
9.                 $headers = trim($requestHeaders["X-jwt-assertion"]);
10.            }
11.            return $headers;
12.        } else {
13.            $headers = $_SERVER["HTTP_X_JWT_ASSERTION"];
14.            $headers = trim($headers);
15.            return $headers;
16.        }
17.    }

```

Kode sumber 4.37 fungsi menangkap token pada header

4.5.6 Implementasi Push Notification

Implementasi fitur *push notification* telah diterapkan dan dapat digunakan untuk mengirimkan pemberitahuan ke pengguna yang mengizinkan fitur tersebut. *Push notification* menerapkan *firebase cloud messaging* untuk web dengan menggunakan bahasa pemrograman JavaScript. Kode sumber 4.38 merupakan konfigurasi yang dibutuhkan untuk dapat mengaplikasikan *firebase cloud messaging*.

```

1. var config = {
2.
3.     apiKey: "AIzaSyA0XZ2KcSNVRUcQbhhkauFB4i-L5bkBnUM",
4.     authDomain: "parabolic-env-189004.firebaseio.com",
5.     databaseURL: "https://parabolic-env-189004.firebaseio.com",
6.     projectId: "parabolic-env-189004",
7.     storageBucket: "parabolic-env-189004.appspot.com",
8.     messagingSenderId: "921732324714"

```

```
9.    };
```

Kode sumber 4.38 Konfigurasi firebase cloud messaging

Setelah konfigurasi tersebut dilakukan, maka selanjutnya menginisialisasi pengguna firebase dengan cara seperti pada Kode sumber 4.39.

```
1.    firebase.initializeApp(config);
```

Kode sumber 4.39 Inisialisasi penggunaan firebase

setelah menginisialisasi penggunaan firebase maka langkah selanjutnya adalah menambahkan script untuk meminta izin kepada *browser* yang ditentukan oleh pengguna, Kode sumber 4.40 merupakan *script* yang diimplementasikan.

```
1.    const messaging = firebase.messaging();
2.    messaging.requestPermission().then(function() {
3.      console.log('Notification permission granted.');
```

```
4.      if (isTokenSentToServer()) {
5.        console.log('Token already saved.');
```

```
6.      } else {
7.        getRegToken();
8.      }
9.    }).catch(function(err) {
10.     console.log('Unable to get permission to notify.', err);
11.    });
```

Kode sumber 4.40 Fungsi untuk meminta izin fitur *push notification*

Fungsi tersebut akan meminta izin fitur diizinkan atau tidak diizinkan, apabila fitur tersebut diizinkan maka fungsi tersebut akan memanggil fungsi untuk membuat sebuah token jika token belum pernah dibuat sama sekali. Kode sumber 4.41 merupakan implementasi script yang dilakukan untuk membuat sebuah token, token tersebut otomatis dibuat oleh firebase.


```

1. function getRegToken(argument){
2.   messaging.getToken().then(function(currentToken) {
3.     if (currentToken) {
4.       saveToken(currentToken);
5.       console.log(currentToken);
6.       setTokenSentToServer(true);
7.     } else {
8.       console.log('No Instance ID token available. Request permission to generate one. ');
9.       setTokenSentToServer(false);
10.    }
11.  }).catch(function(err) {
12.    console.log('An error occurred while retrieving token. ', err);
13.    showToken('Error retrieving Instance ID token. ', err);
14.    setTokenSentToServer(false);
15.  });
16. }

```

Kode sumber 4.41 Fungsi untuk mendapatkan token

Setelah token tersebut dibuat, maka token tersebut akan disimpan ke basis data myITS, selanjutnya jika ingin mengirimkan pemberitahuan kepada pengguna SIAKAD, maka dapat dilakukan dengan menggunakan *tools* postman yaitu mengakses endpoint seperti localhost/siakad/src/push/send. Kode sumber 4.42 merupakan contoh hasil mengirimkan pemberitahuan dengan menggunakan postman.

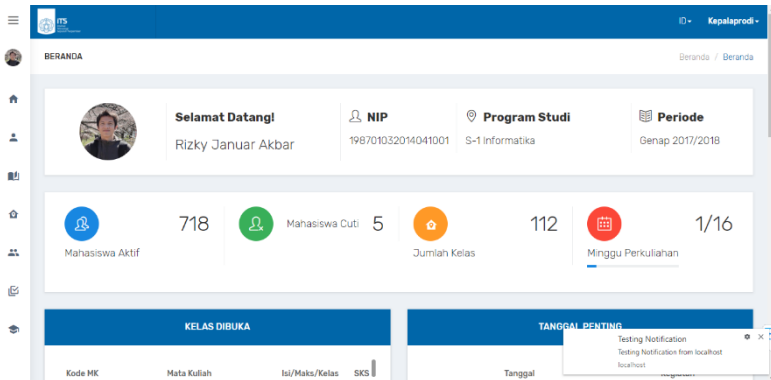
```

1. {"multicast_id":8801938211132552721,"success":1,"failure":0,"canonical_ids":0,"results":[{"message_id":"0:1530864890200878%e609af1cf9fd7ecd"}]}

```

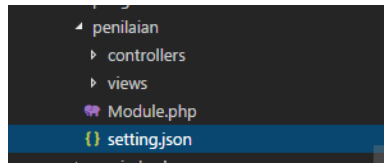
Kode sumber 4.42 status pengiriman *push notification*

Apabila pemberitahuan tersebut sukses untuk dikirimkan maka status pada parameter *success* bernilai 1. Gambar 4.12 merupakan contoh pemberitahuan muncul pada pengguna yang berupa *pop-up*.



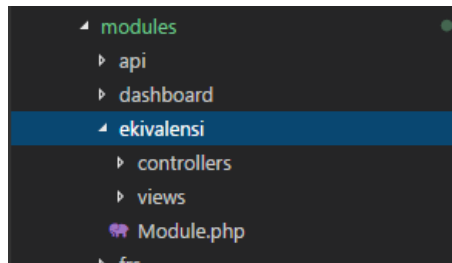
Gambar 4.12 Pesan pop up pada fitur push notification

Modul pengaturan diterapkan untuk mengatur setiap data-data penting yang selalu berubah seiring berjalannya waktu. Dengan adanya modul pengaturan semua modul yang memerlukan data-data penting dapat mengambil data tersebut dalam format JSON, dimana data-data tersebut dapat diambil pada file setting.JSON yang terdapat pada setiap modul. Modul pengaturan dapat diolah oleh pengguna yang berhak untuk mengakses modul tersebut. Pada modul tersebut pengguna hanya dapat melihat informasi penting terkait data-data penting yang terdapat pada setiap modul dan dapat memperbarui informasi dengan memperbarui data-data yang ada dengan mengisi form yang telah disediakan. Pengguna hanya dapat mengolah data-data pada setiap modul jika modul tersebut memiliki pengaturan, jika tidak maka pengguna tidak dapat mengolah dan modul tersebut tidak memiliki file setting.JSON. Gambar 4.13 merupakan contoh modul yang memiliki pengaturan dan memiliki file setting.JSON.



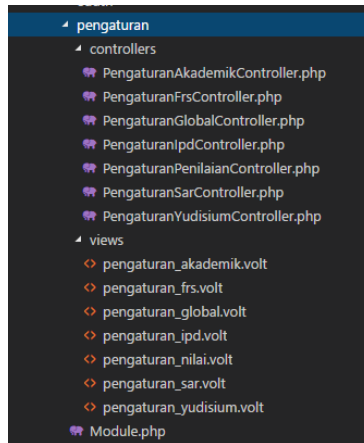
Gambar 4.13 file setting.json

Tidak semua modul memiliki file setting.JSON, hal tersebut dikarenakan modul tersebut belum memiliki kebutuhan data-data yang harus di atur pada modul pengaturan. Gambar 4.14 merupakan contoh modul yang tidak memiliki file setting.JSON pada folder modul tersebut.



Gambar 4.14 modul tanpa file setting.json

Jika modul tidak terdapat file pengaturan, maka pada modul pengaturan tidak tersedia halaman untuk pengaturan modul tersebut. Jika suatu saat modul tersebut harus memiliki file setting.json, maka *programmer* terlebih dahulu harus menambahkan halaman untuk mengolah data pengaturan modul tersebut dan membuat form yang diperlukan pada halaman tersebut, sehingga pengguna dapat mengolah modul tersebut. Gambar 4.15 merupakan struktur direktori folder pada modul pengaturan.



Gambar 4.15 struktur direktori modul pengaturan

Untuk pengembangan modul pengaturan selanjutnya, apabila terdapat penambahan modul yang ingin menerapkan file setting.JSON, maka *programmer* harus membuat *controller* dan *views* pada folder yang tersedia, dan selanjutnya *programmer* membuat form sesuai dengan kebutuhan-kebutuhan data yang diperlukan.

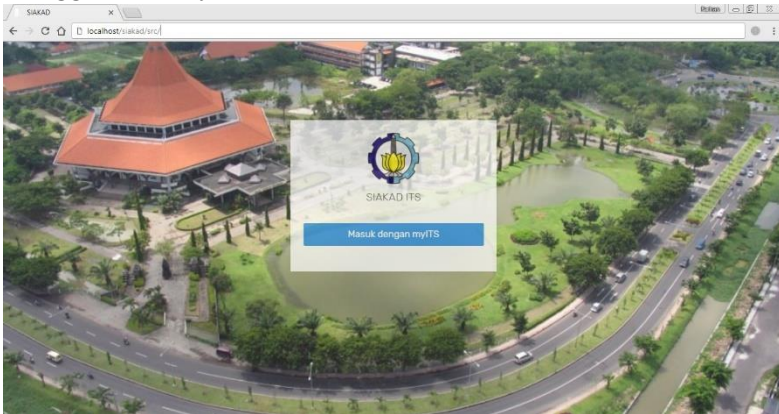
4.6 Implementasi Perancangan Antarmuka

Pada subbab ini akan menjelaskan perancangan antarmuka yang telah diimplementasikan pada Tugas Akhir.

4.6.1 Antarmuka Halaman Login SIAKAD

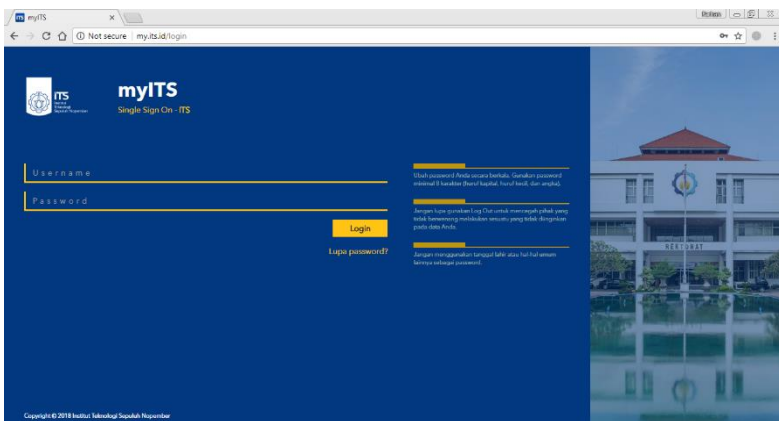
Pada antar muka halaman login SIAKAD hanya terdapat tombol yang akan mengarahkan pengguna untuk dapat login dengan myITS. Pada aplikasi SIAKAD tidak ada form untuk memasukkan username dan password, dikarenakan aplikasi SIAKAD menerapkan sebagai aplikasi klien dari aplikasi myITS,

sehingga untuk dapat masuk ke SIAKAD, pengguna harus login menggunakan myITS.



Gambar 4.16 halaman login SIAKAD

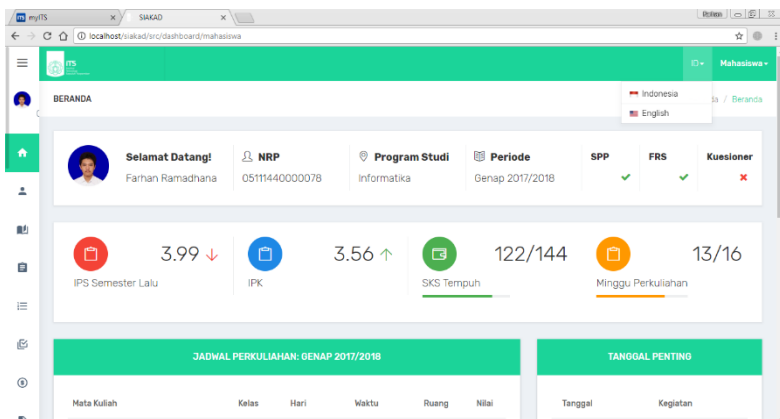
Gambar 4.17 merupakan antar muka halaman login myITS, akan tetapi antar muka tersebut tidak termasuk implementasi pada Tugas Akhir ini melainkan termasuk ke implementasi Tugas Akhir lainnya.



Gambar 4.17 halaman login myITS

4.6.2 Antarmuka Halaman *Dashboard* Untuk Mengganti Bahasa

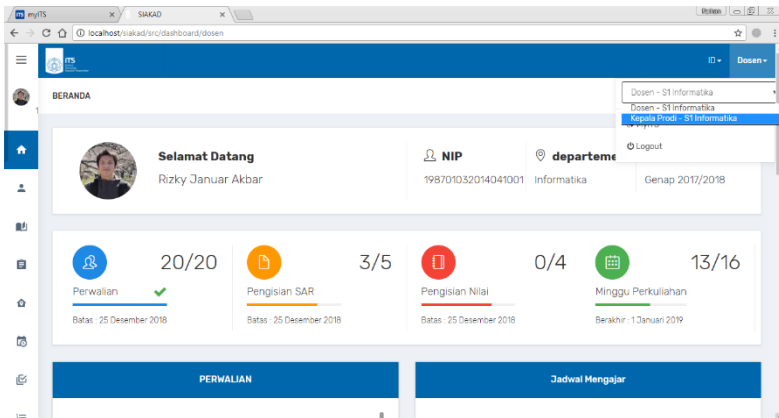
Untuk dapat mengganti bahasa pengantar yang disediakan oleh SIAKAD, pengguna dapat menggantinya pada topbar. Semua pengguna pada SIAKAD dapat menggunakan fitur ini. Semua halaman pada SIAKAD memiliki desain dan perancangan elemen-elemen topbar yang sama sehingga opsi untuk pemilihan bahasa dapat dilakukan disemua halaman dashboard SIAKAD. Gambar 4.18 merupakan gambar antar muka halaman dashborad untuk mengantu bahasa.



Gambar 4.18 antarmuka halaman mengganti bahasa pengantar

4.6.3 Antarmuka Halaman *Dashboard* Untuk Mengganti Hak Akses

Pengguna yang memiliki hak akses lebih dari satu pada SIAKAD maka dapat memilih hak akses pengguna pada topbar SIAKAD. Pengguna dapat mengganti hak akses disemua halaman dashboard SIAKAD. Gambar 4.19 merupakan halaman dashboard untuk mengganti hak akses.



Gambar 4.19 antarmuka halaman mengganti hak akses pengguna

4.6.4 Antarmuka Halaman Modul Pengaturan

Halaman yang akan muncul pertama kali jika pengguna mengakses modul pengaturan adalah halaman yang berisikan informasi daftar-daftar modul yang terdapat pada SIAKAD. Untuk dapat mengakses modul yang diinginkan pada halaman ini, maka pengguna dapat klik tombol yang tersedia, jika tombol tidak bisa di klik maka modul tersebut tidak atau belum memiliki file pengaturan. Berikut merupakan halaman daftar modul yang terdapat pada SIAKAD pada halaman modul pengaturan. Modul pengaturan menyediakan pengaturan umum, dimana informasi pada pengaturan umum tersebut dapat digunakan pada semua modul di SIAKAD. Gambar 4.21 merupakan halaman antar muka modul pengaturan untuk pengaturan umum.

PENGATURAN MODUL	
Nama Modul	File Pengaturan
Global	
Ekivalensi	
Frs	
lpd	

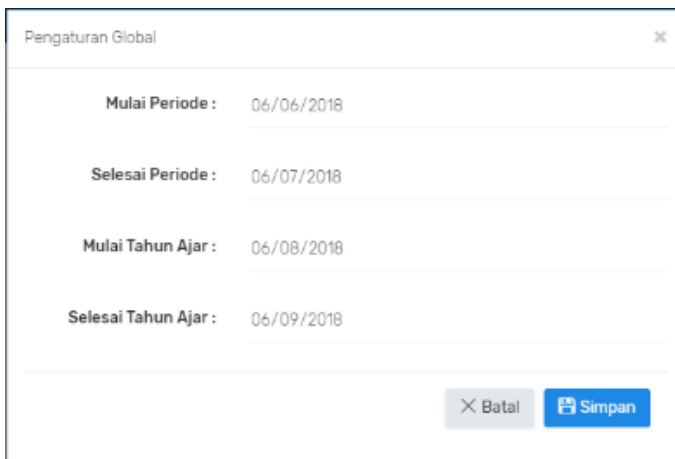
Gambar 4.20 antarmuka halaman modul pengaturan

Informasi Global	
Nama Pengaturan	Nilai Pengaturan
Mulai Periode	2018-06-06
Selesai Periode	2018-06-07
Mulai Tahun Ajar	2018-06-08
Selesai Tahun Ajar	2018-06-09
 	

Gambar 4.21 antarmuka halaman modul pengaturan umum

Informasi pada halaman tersebut dapat diperbarui dengan form yang telah disediakan. Untuk dapat mengubah data pada halaman tersebut maka pengguna harus mengklik tombol yang berwarna biru. Setelah pengguna mengklik tombol tersebut, maka halaman

akan memunculkan modal yang berisikan form. Gambar 4.22 merupakan antar muka halaman untuk mengubah data informasi pada modul pengaturan umum.



The image shows a modal window titled "Pengaturan Global" with a close button (X) in the top right corner. The form contains four rows of input fields, each with a label and a date value:

Label	Value
Mulai Periode :	06/06/2018
Selesai Periode :	06/07/2018
Mulai Tahun Ajar :	06/08/2018
Selesai Tahun Ajar :	06/09/2018

At the bottom right of the modal, there are two buttons: a grey "X Batal" button and a blue "Simpan" button with a save icon.

Gambar 4.22 modal untuk mengubah data modul pengaturan umum

Hal yang sama juga dapat dilakukan pada semua halaman modul pengaturan. Gambar 4.23 merupakan antar muka halaman pengaturan untuk modul FRS.

Informasi FRS	
Nama Pengaturan	Nilai Pengaturan
Mulai FRS	2018-07-01
Selesai FRS	2018-07-02
Batas Ubah FRS	2018-07-03
Batas Drop FRS	2018-07-04

Gambar 4.23 antarmuka halaman modul pengaturan FRS

Modal untuk mengubah informasi pada modul pengaturan FRS.

Pengaturan FRS

Mulai Frs :

07/01/2018

Selesai Frs :

07/02/2018

Batas Ubah Frs :

07/03/2018

Batas Drop Frs :

07/04/2018

X Batal

Simpan

Gambar 4.24 modal untuk mengubah data modul pengaturan FRS

Halaman antar muka modul pengaturan IPD.

Informasi IPD	
Nama Pengaturan	Nilai Pengaturan
Mulai IPD	2018-06-08
Selesai IPD	2018-06-09



Gambar 4.25 antarmuka halaman modul pengaturan IPD

Modal untuk mengubah informasi pada modul pengaturan IPD.

Pengaturan IPD

Mulai IPD :

06/08/2018

Selesai IPD :

06/09/2018

✕ Batal

💾 Simpan

Gambar 4.26 modal untuk mengubah data modul pengaturan IPD

Halaman antar muka modul pengaturan penilaian.

Informasi Penilaian	
Nama Pengaturan	Nilai Pengaturan
Mulai Nilai	2018-06-07
Selesai Nilai	2018-06-08

Gambar 4.27 antarmuka halaman modul pengaturan penilaian

Modal untuk mengubah informasi pada modul pengaturan penilaian.

Pengaturan Nilai

Mulai Nilai :

06/07/2018

Selesai Nilai :

06/08/2018

✕ Batal

Simpan

Gambar 4.28 modal untuk mengubah data modul pengaturan penilaian

Halaman antar muka modul pengaturan SAR.

Informasi SAR	
Nama Pengaturan	Nilai Pengaturan
Mulai SAR	2018-06-01
Selesai SAR	2018-06-02

Gambar 4.29 antarmuka halaman modul pengaturan SAR

Modal untuk mengubah informasi pada modul pengaturan SAR.

Pengaturan SAR

Mulai SAR :

06/01/2018

Selesai SAR :

06/02/2018

X Batal

Simpan

Gambar 4.30 modal untuk mengubah data modul pengaturan SAR

Halaman antar muka modul pengaturan yudisium.

Informasi Yudisium	
Nama Pengaturan	Nilai Pengaturan
Mulai Yudisium	2018-06-15
Selesai Yudisium	2018-06-02



Gambar 4.31 antarmuka halaman modul pengaturan yudisium

Modal untuk mengubah informasi pada modul pengaturan yudisium.

Pengaturan Yudisium ✕

Mulai Yudisium :	06/15/2018
Selesai Yudisium :	06/02/2018

✕ Batal Simpan

Gambar 4.32 modal untuk mengubah data modul pengaturan yudisium

4.7 Implementasi Pada Server

SIAKAD diunggah (*deploy*) ke server. Web server yang digunakan adalah nginx. Untuk dapat mengunggah SIAKAD ke server maka diperlukan konfigurasi pada file siakad yang terdapat pada folder `/etc/nginx/sites-available` seperti pada Kode sumber 4.43.

```

1.  server {
2.      # Port 80 will require Nginx to be started with root permissions
3.      # Depending on how you install Nginx to use port 80 you will need
4.      # to start the server with `sudo` ports about 1000 do not require
5.      # root privileges
6.      # listen    80;
7.
8.      listen    80;
9.      server_name _;
10.
11.     #####
12.     # In production require SSL
13.     # listen 443 ssl default_server;
14.
15.     # ssl on;
16.     # ssl_session_timeout 5m;
17.     # ssl_protocols SSLv2 SSLv3 TLSv1;
18.     # ssl_ciphers ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:
19.     # ssl_prefer_server_ciphers on;
20.
21.     # These locations depend on where you store your certs
22.     # ssl_certificate    /var/nginx/certs/default.cert;
23.     # ssl_certificate_key /var/nginx/certs/default.key;
24.     #####
25.
26.     # This is the folder that index.php is in
27.     root /var/www/siakad;
28.     index index.php index.html index.htm;
29.
30.     charset utf-8;
31.     client_max_body_size 100M;
32.     fastcgi_read_timeout 1800;
33.

```

```

34. # Represents the root of the domain
35. # http://localhost:8000/[index.php]
36. location / {
37.     # Matches URLs `$_GET['_url']`
38.     try_files $uri $uri/ /index.php?_url=$uri&$args;
39. }
40.
41. # When the HTTP request does not match the above
42. # and the file ends in .php
43. location ~ [^/]\.php(/|$) {
44.     # try_files $uri =404;
45.
46.     # Ubuntu and PHP7.0-fpm in socket mode
47.     # This path is dependent on the version of PHP install
48.     fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
49.
50.     # Alternatively you use PHP-
    FPM in TCP mode (Required on Windows)
51.     # You will need to configure FPM to listen on a standard port
52.     # https://www.nginx.com/resources/wiki/start/topics/examples/phpfast
    cgionwindows/
53.     # fastcgi_pass 127.0.0.1:9000;
54.
55.     fastcgi_index /index.php;
56.
57.     include fastcgi_params;
58.     fastcgi_split_path_info ^(.+?\.php)(/.*)$;
59.     if (!-f $document_root$fastcgi_script_name) {
60.         return 404;
61.     }
62.
63.     fastcgi_param PATH_INFO    $fastcgi_path_info;
64.     # fastcgi_param PATH_TRANSLATED $document_root$fastcgi_path
    h_info;
65.     # and set php.ini cgi.fix_pathinfo=0
66.
67.     fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_
    name;
68. }
69.
70. location ~ /\.ht {
71.     deny all;
72. }

```



```

73.
74.     location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
75.         expires      max;
76.         log_not_found off;
77.         access_log    off;
78.     }
79. }

```

Kode sumber 4.43 konfigurasi file SIAKAD di nginx

File tersebut selanjutnya akan di *symlink* ke direktori /etc/nginx/sites-enabled. File web SIAKAD dilokasikan pada /home/farhan seperti Gambar 4.33.

/home/farhan/				
Name	Size	Changed	Rights	Owner
..		5/14/2018 2:03:18 PM	rwxf-r-x	root
siakad		5/23/2018 11:37:30 PM	rwxf-r-x	farhan

Gambar 4.33 folder SIAKAD di server

Selanjutnya, file tersebut akan di *symlink* ke /var/www/siakad agar dapat diakses melalui url. Gambar 4.34 merupakan isi dari direktori /var/www/siakad.

/var/www/siakad/				
Name	Size	Changed	Rights	Owner
..		5/15/2018 6:24:47 PM	rwxf-r-x	farhan
upload		5/16/2018 3:23:26 PM	rwxf-rwx	farhan
plugins		5/16/2018 3:23:15 PM	rwxf-rwx	farhan
less		5/16/2018 3:22:59 PM	rwxf-rwx	farhan
js		5/16/2018 3:22:49 PM	rwxf-rwx	farhan
files		5/16/2018 3:22:39 PM	rwxf-rwx	farhan
css		5/16/2018 3:22:21 PM	rwxf-rwx	farhan
bootstrap		5/16/2018 12:46:11 PM	rwxf-rwx	farhan
index.php	1 KB	5/16/2018 12:59:58 PM	rwxf-rwx	farhan

Gambar 4.34 folder SIAKAD di *symlink* ke /var/www/siakad

[halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari kerangka kerja Sistem Informasi Akademik (SIKAD).

5.1 Lingkungan Uji Coba

Lingkungan uji coba adalah kombinasi antara perangkat keras dan perangkat lunak yang digunakan untuk melakukan uji coba. Pengujian dilakukan dengan lingkungan pengujian pada klien. Adapun rincian spesifikasi lingkungan pengujian tersebut ditunjukkan pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Klien

Spesifikasi	Deskripsi
CPU	Intel® Core™ i7-4500U CPU @ 1.80GHz
RAM	8.00 GB
Sistem Operasi	Windows 10 Education 64-bit
Browser	Google Chrome

5.2 Skenario Pengujian

Pada bagian ini membahas mengenai proses uji coba. Pengujian dilakukan dengan metode *black box* untuk menguji masing-masing fungsionalitas yang terdapat pada sistem. Metode *black box* adalah metode pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan tanpa memandang struktur internalnya.

Pada proses uji coba, pengujian akan dilakukan dengan memasukkan data uji yang memiliki nilai benar dan salah dan akan melihat hasil yang akan dihasilkan oleh aplikasi.

5.2.1 Kasus Pengujian Oauth2

Kasus pengujian ini dilakukan untuk memeriksa apakah sistem sudah mampu melakukan proses autentikasi dan otorisasi. Dalam kasus pengujian ini, pengguna pada sistem akan mencoba untuk memasuki ke dalam sistem dengan akun yang telah terdaftar pada sistem. Untuk dapat membuktikan Oauth2 dapat berjalan dengan baik adalah pengujian akan dilakukan dengan memasukkan data dari pengguna, yaitu data yang valid dan data yang tidak valid. Detil dari kasus pengujian ini dijelaskan pada Tabel 5.2.

Tabel 5.2 Kasus Pengujian Oauth2

ID	UJ-001
Fitur	OAuth2
Nama	Pengujian Oauth2
Tujuan Pengujian	Menguji apakah sistem sudah mampu menerapkan Oauth2 secara baik dan benar
Skenario 1	Pengguna masuk dengan data yang sesuai
Kondisi Awal	Pengguna belum <i>login</i> ke dalam sistem
Data Uji	Username: farhan Password: lalala
Langkah Pengujian	1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang sesuai atau data benar 2. Pengguna menekan tombol masuk
Hasil yang diharapkan	Pengguna dapat masuk ke <i>dashboard</i> SIAKAD dan sesuai dengan hak akses pengguna

Kondisi Akhir	Sistem mengarahkan pengguna ke halaman <i>dashboard</i> sesuai dengan hak akses pengguna
Skenario 2	Pengguna masuk dengan data yang tidak sesuai
Kondisi Awal	Pengguna belum <i>login</i> ke dalam sistem
Data Uji	Username: farhan Password: cacaca
Langkah Pengujian	1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang tidak sesuai atau data salah 2. Pengguna menekan tombol masuk
Hasil Yang Diharapkan	Pengguna tidak dapat masuk ke halaman <i>dashboard</i> SIAKAD
Kondisi Akhir	Sistem mengarahkan pengguna ke halaman <i>login</i> dan memberitahukan pengguna untuk memasukkan <i>username</i> dan <i>password</i> yang sesuai

5.2.2 Kasus Pengujian RBAC

Kasus pengujian ini dilakukan untuk memeriksa apakah sistem sudah mampu menentukan RBAC sesuai dengan data yang dimasukkan oleh pengguna. Dalam kasus pengujian ini, pengguna pada sistem akan mencoba untuk memasuki ke dalam sistem dengan akun yang telah terdaftar pada sistem. Untuk dapat membuktikan RBAC dapat berjalan dengan baik adalah pengujian akan dilakukan dengan memasukkan data dari pengguna dengan beberapa data pengguna yang memiliki perbedaan hak akses. Detil dari kasus pengujian ini dijelaskan pada Tabel 5.3.

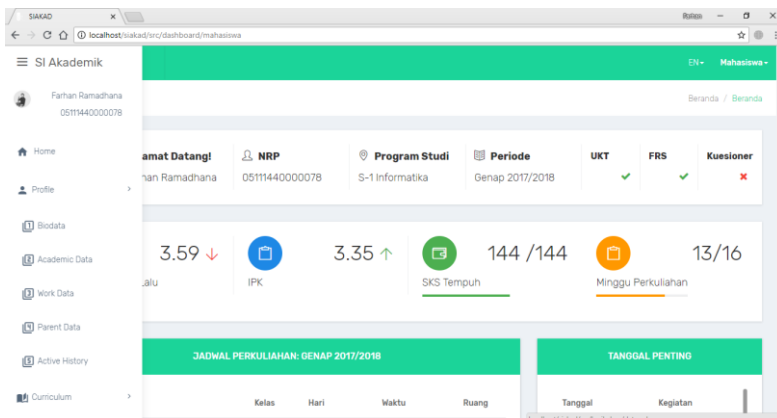
Tabel 5.3 Kasus Pengujian RBAC

ID	UJ-002
Fitur	RBAC
Nama	Pengujian RBAC
Tujuan Pengujian	Untuk menguji apakah pengguna mendapatkan hak akses yang sesuai atau tidak
Skenario 1	Pengguna merupakan mahasiswa
Kondisi Awal	Pengguna belum masuk ke SIAKAD. Pengguna sudah berada pada halaman login di myITS
Data Uji	Username: farhan Password: lalala
Langkah Pengujian	1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang benar 2. Pengguna menekan tombol masuk

Hasil yang diharapkan	Pengguna masuk dan diarahkan ke <i>dashboard</i> mahasiswa, serta mendapatkan menu-menu sesuai dengan hak akses mahasiswa
Kondisi Akhir	Pengguna diarahkan dan mendapatkan menu-menu sesuai dengan hak akses mahasiswa
Skenario 2	Pengguna merupakan dosen
Kondisi Awal	Pengguna belum masuk ke SIAKAD. Pengguna sudah berada pada halaman <i>login</i> di myITS
Data Uji	Username: rizkyja Password: lalala
Langkah Pengujian	1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang benar 2. Pengguna menekan tombol masuk
Hasil yang diharapkan	Pengguna masuk dan diarahkan ke <i>dashboard</i> dosen, serta mendapatkan menu-menu sesuai dengan hak akses dosen
Kondisi Akhir	Pengguna diarahkan dan mendapatkan menu-menu sesuai dengan hak akses dosen
Skenario 3	Pengguna merupakan dosen dan juga ketua RMK
Kondisi Awal	Pengguna belum masuk ke SIAKAD. Pengguna sudah berada pada halaman <i>login</i> di myITS

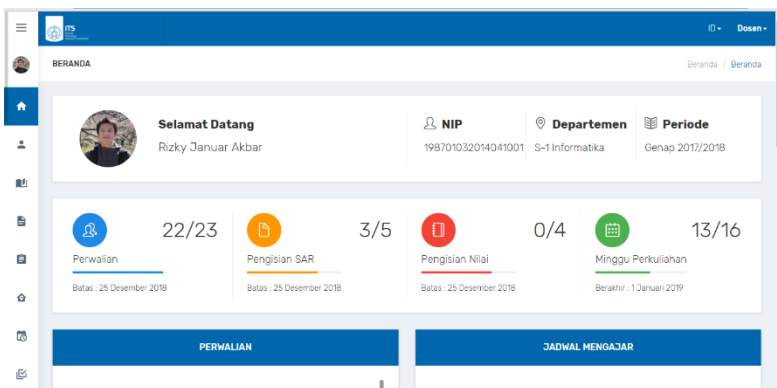
Data Uji	Username: sitirochimah Password: lalala
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang benar 2. Pengguna menekan tombol masuk
Hasil yang diharapkan	Pengguna masuk dan diarahkan ke <i>dashboard</i> ketua RMK serta mendapatkan menu-menu sesuai dengan hak akses ketua RMK, serta pengguna dapat mengganti hak akses sebagai dosen dan mendapatkan menu-menu sesuai hak akses dosen
Kondisi Akhir	Pengguna diarahkan dan mendapatkan menu-menu sesuai dengan hak akses ketua RMK dan dosen

Hasil pengujian yang dilakukan dengan memasukkan data pengguna yang memiliki hak akses mahasiswa (skenario 1).



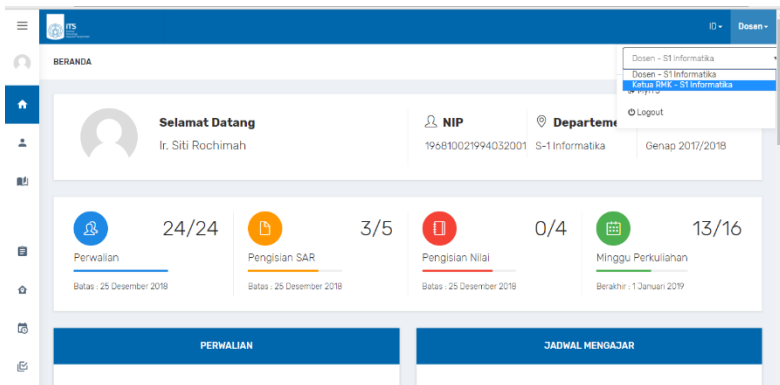
Gambar 5.1 hasil pengujian skenario 1 pada kasus pengujian RBAC

Hasil pengujian yang dilakukan dengan memasukkan data pengguna yang memiliki hak akses dosen (skenario 2).



Gambar 5.2 hasil pengujian skenario 2 pada kasus pengujian RBAC

Hasil pengujian yang dilakukan dengan memasukkan data pengguna yang memiliki hak akses dosen dan ketua RMK (skenario 3).



Gambar 5.3 hasil pengujian skenario 3 pada kasus pengujian RBAC

5.2.3 Kasus Pengujian Internationalization

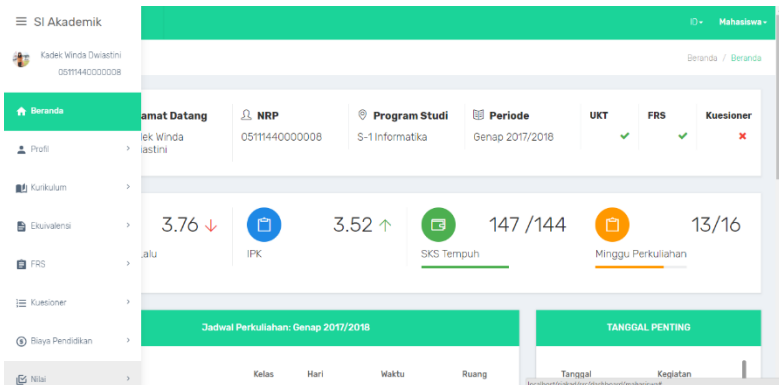
Kasus pengujian ini dilakukan untuk memeriksa dan menguji apakah sistem sudah dapat menerapkan fitur *internationalization* dengan baik. Dalam kasus ini skenario pengujian akan dibagi menjadi 3 skenario dimana skenario pertama akan dilakukan dengan data uji pengguna yang memiliki nilai locale ID pada informasi yang didapatkan dari endpoint /userinfo pada aplikasi myITS, skenario kedua akan dilakukan dengan data uji pengguna yang memiliki nilai locale EN, dan pengujian juga akan dilakukan dengan mengganti bahasa yang tersedia pada topbar SIAKAD. Detil dari kasus pengujian ini dijelaskan pada Tabel 5.4.

Tabel 5.4 Kasus pengujian *internationalization*

ID	UJ-003
Fitur	Internationalization
Nama	Pengujian Internationalization
Tujuan Pengujian	Untuk menguji apakah pengguna mendapatkan pilihan bahasa yang sesuai dengan informasi locale dan apakah pengguna dapat mengganti bahasa yang digunakan
Skenario 1	
Kondisi Awal	Pengguna masuk sebagai mahasiswa yang memiliki informasi nilai locale berupa ID
Data Uji	Username: windadwiasitini Password: lalala
Langkah Pengujian	1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang benar 2. Pengguna menekan tombol masuk
Hasil yang diharapkan	Pengguna masuk dan diarahkan ke dashboard mahasiswa dengan bahasa Indonesia
Kondisi Akhir	Pengguna masuk ke dashboard mahasiswa dan menggunakan bahasa Indonesia
Skenario 2	
Kondisi Awal	Pengguna masuk sebagai mahasiswa yang memiliki

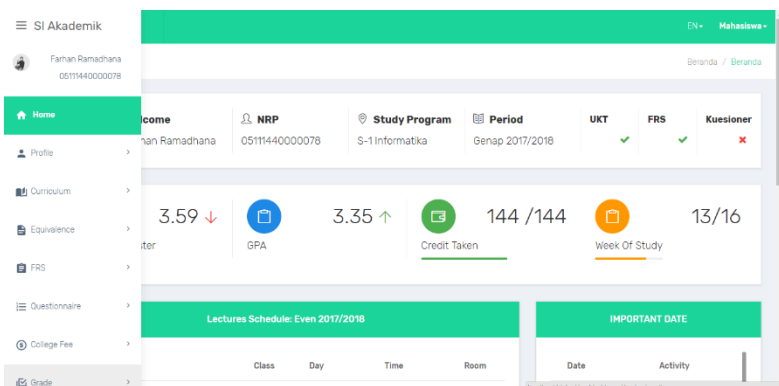
	informasi nilai locale berupa EN
Data Uji	Username: farhan Password: lalala
Langkah Pengujian	1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang benar 2. Pengguna menekan tombol masuk
Hasil yang diharapkan	Pengguna masuk dan diarahkan ke dashboard mahasiswa dengan bahasa Inggris
Kondisi Akhir	Pengguna masuk ke dashboard mahasiswa dan menggunakan bahasa Inggris
Skenario 3	
Kondisi Awal	Pengguna sudah masuk ke SIAKAD pada dashboard dosen dengan bahasa Indonesia.
Data Uji	Username: rizkyja Password: lalala
Langkah Pengujian	1. Pengguna mengubah bahasa pengantar menjadi bahasa Inggris pada topbar
Hasil yang diharapkan	Pengguna dapat mengubah bahasa pengantar dari bahasa Indonesia ke bahasa Inggris
Kondisi Akhir	Bahasa pengantar pada dashboard berubah menjadi bahasa Inggris

Hasil pengujian yang dilakukan dengan memasukkan data pengguna yang memiliki informasi *locale* ID (skenario 1).



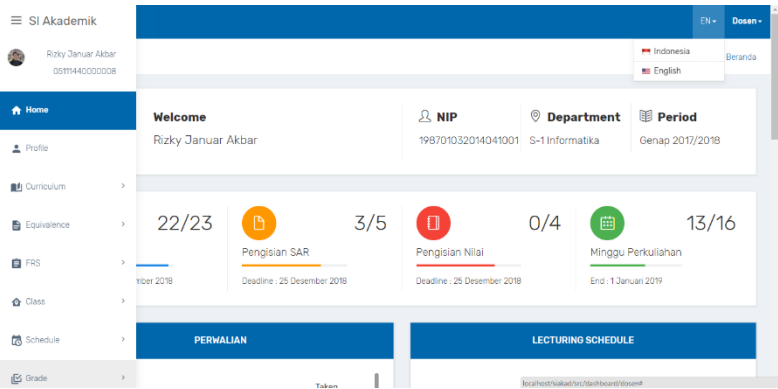
Gambar 5.4 hasil pengujian skenario 1 pada kasus pengujian *internationalization*

Hasil pengujian yang dilakukan dengan memasukkan data pengguna yang memiliki informasi *locale* EN (skenario 2).



Gambar 5.5 hasil pengujian skenario 2 pada kasus pengujian *internationalization*

Hasil pengujian yang dilakukan dengan memasukkan data pengguna yang memiliki informasi *locale* ID dan mengganti bahasa ke bahasa Inggris (skenario 3).



Gambar 5.6 hasil pengujian skenario 2 pada kasus pengujian *internationalization*

5.2.4 Kasus Pengujian Logging

Kasus pengujian ini dilakukan untuk memeriksa apakah pencatatan log pada SIAKAD sudah berhasil diterapkan atau tidak. Pencatatan log berhasil jika log dapat disimpan pada basis data aplikasi myITS. Dalam kasus ini, pengujian akan dilakukan pada 2 skenario, dimana disetiap skenario tersebut pencatatn log akan dilakukan. Detil dari pengujian logging dapat dilihat pada Tabel 5.5.

Tabel 5.5 Kasus pengujian logging

ID	UJ-004
Fitur	Logging
Nama	Pengujian logging
Tujuan Pengujian	Untuk menguji apakah sistem sudah mencatat log dengan

	baik dan menyimpan ke basis data myITS
Skenario 1	
Kondisi Awal	Pengguna belum masuk ke aplikasi SIAKAD, pengguna sudah berada di halaman <i>login</i> myITS untuk masuk ke SIAKAD
Data Uji	Username: farhan Password: lalala
Langkah Pengujian	1. pengguna memasukkan username dan password berdasarkan data uji pada form login 2. pengguna menekan tombol masuk
Hasil yang diharapkan	Log akan dicatat dan disimpan ke basis data myITS, yaitu log dengan kategori 1 (kategori ini ditentukan untuk kategori login)
Kondisi Akhir	Log berhasil dicatat dan disimpan ke basis data myITS dengan aktivitas “berhasil masuk ke aplikasi SIAKAD dengan sebagai mahasiswa”
Skenario 2	
Kondisi Awal	Pengguna sudah masuk ke SIAKAD sebagai mahasiswa dan berada di dashboard mahasiswa
Data Uji	Username: farhan Password: lalala

Langkah Pengujian	1. pengguna memilih menu melihat nilai kuliah per semester
Hasil Yang Diharapkan	Log akan dicatat dan disimpan ke basis data myITS, yaitu log dengan kategori 11 (kategori ini ditentukan untuk kategori info)
Kondisi Akhir	Log berhasil dicatat dan disimpan ke basis data myITS dengan aktivitas “berhasil masuk ke halaman laporan nilai mahasiswa”

5.2.5 Kasus Pengujian API

Kasus pengujian ini dilakukan untuk memeriksa mekanisme API yang telah disediakan telah berjalan dengan baik. Pada kasus ini pengujian dilakukan dengan bantuan *tools* postman untuk mengirimkan data ke endpoint API. Detil dari kasus pengujian ini seperti Tabel 5.6.

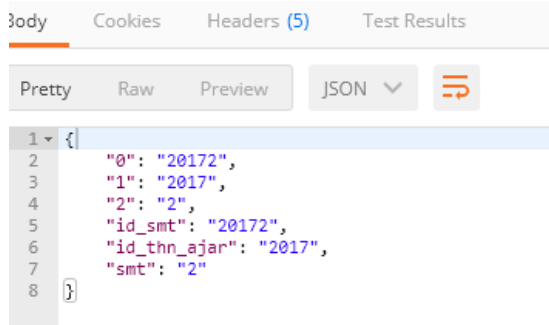
Tabel 5.6 Kasus pengujian API

ID	UJ-005
Fitur	API
Nama	Pengujian API
Tujuan Pengujian	Untuk memeriksa apakah mekanisme API sudah berjalan dengan baik atau tidak
Skenario 1	
Kondisi Awal	Pengguna sudah memiliki token berupa JWT dan menggunakan aplikasi Postman
Data Uji	Header: x-jwt-assertion:

	<p>JWT TOKEN</p> <p>endpoint: https://dev-siakad.its.ac.id/api/v1/penjadwalan/semesterAktif</p>
Langkah Pengujian	<ol style="list-style-type: none"> 1. pengguna memasukkan endpoint API ke postman dengan metode GET 2. pengguna memasukkan data header pada postman
Hasil yang diharapkan	Postman akan menampilkan data yang didapatkan pada aplikasi SIAKAD dalam bentuk JSON
Kondisi Akhir	Data pada SIAKAD dikirimkan dan ditampilkan pada postman
Skenario 2	
Kondisi Awal	Pengguna sudah memiliki token berupa JWT dan menggunakan aplikasi Postman
Data Uji	<p>Header: x-jwt-assertion:</p> <p>JWT TOKEN</p> <p>endpoint: https://dev-siakad.its.ac.id/api/v1/penjadwalan/semesterAktif</p>
Langkah Pengujian	<ol style="list-style-type: none"> 1. pengguna memasukkan endpoint API ke postman dengan metode POST 2. pengguna memasukkan data header pada postman

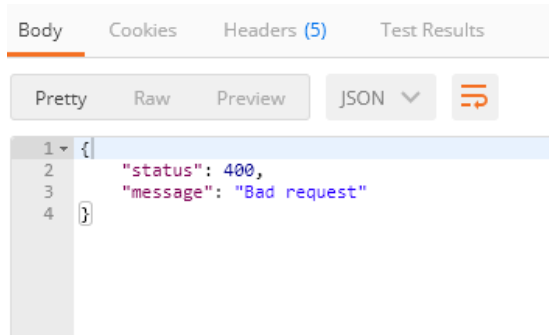
Hasil yang diharapkan	Postman tidak akan menampilkan data karena metode yang digunakan salah
Kondisi Akhir	Status 400 bad request ditampilkan

Hasil dari pengujian API pada skenario 1 adalah seperti Gambar 5.7.



Gambar 5.7 Hasil pengujian skenario 1 pada API

Hasil dari pengujian API pada skenario 2 adalah seperti Gambar 5.8.



Gambar 5.8 Hasil pengujian skenario 2 pada API

5.2.6 Kasus Pengujian Push Notification

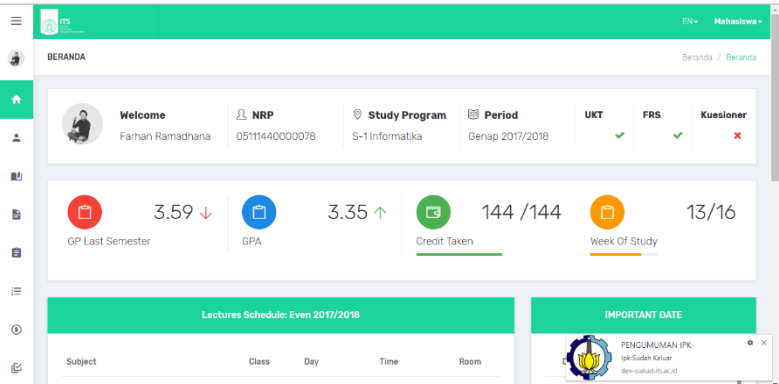
Kasus pengujian ini dilakukan untuk memeriksa fitur *push notification* apakah dapat berjalan dengan baik atau tidak. Dalam kasus pengujian ini, pengguna akan memakai *tools* postman untuk mengirimkan pesan pemberitahuan ke pada pengguna yang telah mengizinkan fitur *push notification*. Detil dari kasus pengujian ini seperti pada Tabel 5.7.

Tabel 5.7 Kasus pengujian pada fitur *push notification*

ID	UJ-006
Fitur	<i>Push notification</i>
Nama	Pengujian <i>Push notification</i>
Tujuan Pengujian	Untuk menguji apakah fitur <i>push notification</i> sudah berjalan dengan baik.
Skenario 1	
Kondisi Awal	Pengguna sudah memiliki akses key untuk mengakses API SIAKAD dan memiliki push key untuk mengakses firebase cloud messaging
Data Uji	<p>Authorization: 61634d8c997bee19368d1ee8a53d53e982409812</p> <p>Push-Key: AlzaSyApBLTniEYZZgkdhFMYj5ZWgE9oujhEA_E</p> <p>Body: { "tittle" : "PENGUMUMAN IPK", "body" : "Ipk Sudah Keluar", "icon": "https://katamata.files.wordpress.com/2011/12/lambang-its-png-v1.png" }</p>
Langkah Pengujian	1. Pengguna memasukkan data header pada aplikasi postman

	2. Pengguna memasukkan data body pada aplikasi postman 3. Pengguna menekan tombol send
Hasil yang diharapkan	Pesan akan dikirimkan ke setiap pengguna dan dapat ditampilkan di perangkat pengguna
Kondisi Akhir	Pesan berhasil dikirimkan dan ditampilkan berupa pop-up ke perangkat pengguna

Hasil pengujian pada skenario 1 pada fitur *push notification* seperti Gambar 5.9.



Gambar 5.9 Hasil pengujian skenario 1 pada fitur *push notification*

5.2.7 Kasus Pengujian Modul Pengaturan

Kasus pengujian ini dilakukan untuk memeriksa modul pengaturan apakah dapat berjalan dengan baik atau tidak. Dalam kasus pengujian ini, pengguna akan mengubah atau memperbarui informasi pada modul pengaturan. Pada modul pengaturan terdapat beberapa file konfigurasi yaitu *setting.json* yang terdapat pada setiap modul. Terdapat 7 skenario pada kasus pengujian ini seperti pada Tabel 5.8.

Tabel 5.8 Kasus pengujian modul pengaturan

ID	UJ-007
Fitur	Modul Pengaturan
Nama	Pengujian modul pengaturan
Tujuan Pengujian	Untuk memeriksa apakah modul pengaturan dapat berjalan dengan baik
Skenario 1	
Kondisi Awal	Pengguna telah masuk ke SIAKAD dengan hak akses sebagai biro akademik
Data Uji	Mulai periode: 06/06/2018, Selesai periode: 06/07/2018, Mulai tahun ajar: 06/08/2018, Selesai tahun ajar: 06/09/2018, Perkuliahan minggu 1: 07/20/2018, Perkuliahan minggu 2: 07/21/2018, Perkuliahan minggu 3: 07/22/2018, Perkuliahan minggu 4: 07/23/2018, Perkuliahan minggu 5: 07/24/2018, Perkuliahan minggu 6: 07/25/2018, Perkuliahan minggu 7: 07/26/2018, Perkuliahan minggu 8: 07/27/2018, Perkuliahan minggu 9: 07/28/2018,

	Perkuliahan minggu 10: 07/29/2018, Perkuliahan minggu 11: 07/30/2018, Perkuliahan minggu 12: 07/31/2018, Perkuliahan minggu 13: 08/01/2018, Perkuliahan minggu 14: 08/02/2018, Perkuliahan minggu 15: 08/03/2018, Perkuliahan minggu 16: 08/04/2018,
Langkah Pengujian	1. Pengguna menekan tombol mengubah pada modul pengaturan global 2. Pengguna mengisi form yang disediakan dengan data uji 3. Pengguna menyimpan data yang telah diisikan pada form
Hasil yang diharapkan	Data yang dimasukkan oleh pengguna dapat disimpan ke dalam file setting.json sesuai dengan modul yang dipilih
Kondisi Akhir	Data berhasil disimpan ke dalam file setting.json pada folder config
Skenario 2	
Kondisi Awal	Pengguna telah masuk ke SIAKAD dengan hak akses sebagai biro akademik
Data Uji	Mulai ekivalensi: 07/01/2018,

	Selesai ekivalensi: 07/02/2018
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol mengubah pada modul pengaturan ekivalensi 2. Pengguna mengisi form yang disediakan dengan data uji 3. Pengguna menyimpan data yang telah diisikan pada form
Hasil yang diharapkan	Data yang dimasukkan oleh pengguna dapat disimpan ke dalam file setting.json sesuai dengan modul yang dipilih
Kondisi Akhir	Data berhasil disimpan ke dalam file setting.json pada folder modul ekivalensi
Skenario 3	
Kondisi Awal	Pengguna telah masuk ke SIAKAD dengan hak akses sebagai biro akademik
Data Uji	Mulai FRS: 07/01/2018, Selesai FRS: 07/02/2018, Batas ubah FRS: 07/03/2018, Batas drop FRS: 07/05/2018
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol mengubah pada modul pengaturan FRS 2. Pengguna mengisi form yang disediakan dengan data uji 3. Pengguna menyimpan data yang telah diisikan pada form
Hasil yang diharapkan	Data yang dimasukkan oleh pengguna dapat disimpan ke

	dalam file setting.json sesuai dengan modul yang dipilih
Kondisi Akhir	Data berhasil disimpan ke dalam file setting.json pada folder modul FRS
Skenario 4	
Kondisi Awal	Pengguna telah masuk ke SIAKAD dengan hak akses sebagai biro akademik
Data Uji	Mulai IPD: 06/08/2018, Selesai IPD: 06/09/2018
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol mengubah pada modul pengaturan IPD 2. Pengguna mengisi form yang disediakan dengan data uji 3. Pengguna menyimpan data yang telah diisikan pada form
Hasil yang diharapkan	Data yang dimasukkan oleh pengguna dapat disimpan ke dalam file setting.json sesuai dengan modul yang dipilih
Kondisi Akhir	Data berhasil disimpan ke dalam file setting.json pada folder modul IPD
Skenario 5	
Kondisi Awal	Pengguna telah masuk ke SIAKAD dengan hak akses sebagai biro akademik
Data Uji	Mulai nilai: 06/07/2018, Selesai nilai: 06/08/2018
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol mengubah pada modul pengaturan penilaian

	2. Pengguna mengisi form yang disediakan dengan data uji 3. Pengguna menyimpan data yang telah diisikan pada form
Hasil yang diharapkan	Data yang dimasukkan oleh pengguna dapat disimpan ke dalam file setting.json sesuai dengan modul yang dipilih
Kondisi Akhir	Data berhasil disimpan ke dalam file setting.json pada folder modul penilaian
Skenario 6	
Kondisi Awal	Pengguna telah masuk ke SIAKAD dengan hak akses sebagai biro akademik
Data Uji	Mulai SAR: 06/01/2018, Selesai SAR: 06/02/2018
Langkah Pengujian	1. Pengguna menekan tombol mengubah pada modul pengaturan SAR 2. Pengguna mengisi form yang disediakan dengan data uji 3. Pengguna menyimpan data yang telah diisikan pada form
Hasil yang diharapkan	Data yang dimasukkan oleh pengguna dapat disimpan ke dalam file setting.json sesuai dengan modul yang dipilih
Kondisi Akhir	Data berhasil disimpan ke dalam file setting.json pada folder modul SAR
Skenario 7	

Kondisi Awal	Pengguna telah masuk ke SIAKAD dengan hak akses sebagai biro akademik
Data Uji	Mulai yudisium: 06/15/2018, Selesai yudisium: 06/02/2018
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol mengubah pada modul pengaturan SAR 2. Pengguna mengisi form yang disediakan dengan data uji 3. Pengguna menyimpan data yang telah diisikan pada form
Hasil yang diharapkan	Data yang dimasukkan oleh pengguna dapat disimpan ke dalam file setting.json sesuai dengan modul yang dipilih
Kondisi Akhir	Data berhasil disimpan ke dalam file setting.json pada folder modul yudisium

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang dapat diambil dari hasil uji coba dan perancangan perangkat lunak sebagai jawaban dari rumusan masalah yang telah dikemukakan dan saran yang berisi pengembangan yang dapat dilakukan lebih lanjut untuk menyempurnakan perangkat lunak.

6.1 Kesimpulan

Berikut merupakan kesimpulan yang dapat diambil dari proses pengembangan dan hasil uji coba.

1. Tahap analisa proses bisnis dan kebutuhan fitur-fitur baru pada SIAKAD telah dilakukan sehingga diambil suatu kesimpulan untuk membangun ulang SIAKAD modular dengan *Framework* Phalcon serta membangun SIAKAD dengan layer integrasi, yang dikerjakan pada Tugas Akhir ini, seperti OAuth2, RBAC, internationalization, *logging* dan *push notification*.
2. SIAKAD yang dibangun telah berhasil menerapkan perancangan yang modular dengan menggunakan arsitektur *Hierarchical Model-View-Controller* dan menggunakan pattern *repository service pattern*.
3. Sistem yang dibangun telah berhasil menerapkan RBAC (*Role Based Access Control*) untuk menentukan dan membatasi hak akses pengguna, untuk menerapkan RBAC maka aplikasi SIAKAD dibangun sebagai aplikasi klien dari aplikasi myITS. Pada aplikasi myITS semua hak akses akan ditentukan.
4. Sistem telah berhasil menerapkan JSON Web Token dan OAuth2 dengan cara menerapkan SIAKAD menjadi aplikasi klien pada myITS, dimana pada myITS untuk

proses otorisasi telah menggunakan standar keamanan OAuth2.

5. Untuk penerapan routing, SIAKAD menggunakan `base_url` yaitu <https://dev-siakad.its.ac.id>. Untuk mengakses sebuah modul dan controller maka format yang digunakan adalah <https://dev-siakad.its.ac.id/modul/controller>.
6. Sistem yang dibangun telah berhasil menerapkan *push notification* sebagai fitur untuk dapat mengirimkan pesan pemberitahuan kepada pengguna dengan menggunakan *firebase cloud messaging*, API sebagai fitur untuk berbagi data dengan sistem yang lainnya, *logging* sebagai fitur untuk mekanisme pencatatan log yang dilakukan oleh pengguna terhadap SIAKAD dengan memanfaatkan kelas `Phalcon\Logger\AdapterInterface`, *internationalization* sebagai fitur untuk mekanisme menerjemahkan bahasa dengan memanfaatkan kelas `Phalcon\Translate\Adapter\NativeArray`.

6.2 Saran

Berikut ini merupakan pengembangan lebih lanjut yang dapat dilakukan untuk menyempurnakan perangkat lunak.

1. Adanya *scheduler* yang berguna untuk melakukan *update* data secara berkala, seperti *update* IPK setiap berapa jam sekali.
2. *logout* pada SIAKAD masih harus dikembangkan lebih lanjut dikarenakan tombol tersebut belum sepenuhnya berfungsi untuk *logout* (hanya mengarahkan pengguna ke halaman masuk SIAKAD), akan tetapi jika pengguna masuk kembali ke sistem pengguna dapat masuk tanpa *login*, dimana seharusnya ketika pengguna sudah *logout* untuk dapat masuk ke aplikasi selanjutnya pengguna harus melakukan *login* kembali.

DAFTAR PUSTAKA

- [1] W. S. Agung, "Rancang Bangun Kerangka Kerja Sistem Informasi Akademik Modular Berbasis WEB Dengan Pola Arsitektur Hierarchical Model-View-Controller," Surabaya, Institut Teknologi Sepuluh Nopember, 2015, p. 1.
- [2] "www.tutorialspoint.com," [Online]. Available: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm. [Accessed 24 3 2018].
- [3] "code.tutsplus.com," [Online]. Available: <https://code.tutsplus.com/tutorials/hmvc-an-introduction-and-application--net-11850>. [Accessed 24 03 2018].
- [4] "azware.wordpress.com," [Online]. Available: <https://azware.wordpress.com/2013/04/06/hmvc-design-pattern-tidak-dapat-menggantikan-source-code-management/>. [Accessed 25 03 2018].
- [5] "coding4ever.net," [Online]. Available: <http://coding4ever.net/blog/2016/02/22/repository-pattern/>. [Accessed 17 04 2018].
- [6] "http://php.net," [Online]. Available: <http://php.net/manual/en/intro-whatis.php>. [Accessed 17 04 2018].
- [7] "http://php.net," [Online]. Available: <http://php.net/manual/en/intro-whatcando.php>. [Accessed 17 04 2018].

- [8] "<https://www.sitepoint.com/>," [Online]. Available: <https://www.sitepoint.com/best-php-frameworks-2014/>. [Accessed 17 04 2018].
- [9] "<https://id.wikipedia.org/>," [Online]. Available: https://id.wikipedia.org/wiki/Microsoft_SQL_Server. [Accessed 17 04 2018].
- [10] R. Mufrizal, "rizkimufrizal.github.io/," [Online]. Available: <https://rizkimufrizal.github.io/belajar-oauth2/>. [Accessed 24 December 2017].
- [11] "<http://searchsecurity.techtarget.com/>," [Online]. Available: <http://searchsecurity.techtarget.com/definition/rolebased-access-control-RBAC>. [Accessed 2018 04 17].
- [12] "<http://www.tutorialspoint.com/>," [Online]. Available: http://www.tutorialspoint.com/restful/restful_introduction.htm. [Accessed 18 04 2018].
- [13] "<https://medium.com/>," [Online]. Available: <https://medium.com/@ahmad.fight/perbedaan-rest-dengan-restfull-api-c08025d6d59e>. [Accessed 18 04 2018].
- [14] "forlap.ristekdikti.go.id/," [Online]. Available: <https://forlap.ristekdikti.go.id/perguruantinggi/homegraphp>t. [Accessed 4 January 2018].

BIODATA PENULIS



Penulis lahir di Banda Aceh, 16 Januari 1997. Penulis telah menempuh pendidikan dasar di SD Negeri 16 Banda Aceh, SMP Negeri 6 Banda Aceh, dan SMA Negeri 10 Fajar Harapan Banda Aceh. Sejak duduk di bangku SMP, penulis suka dengan hal-hal terkait perkembangan teknologi dan perkembangan komputer. Hal tersebut juga yang mendasari penulis melanjutkan pendidikan sarjana S1 di Jurusan Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif berorganisasi, pernah menjadi staf Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS 2015/2016, staf 3D Schematics ITS 2015/2016, Staf Section ITS EXPO 2015/2016 dan Staf Ahli Section ITS EXPO 2015/2016. Penulis juga aktif pada organisasi Pelajar Mahasiswa Kekeluargaan Tanah Rencong (PMKTR) sebagai ketua departemen hubungan luar 2016/2017 dan aktif sebagai administrator laboratorium Rekayasa Perangkat Lunak.

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Rekayasa Perangkat Lunak serta memiliki ketertarikan di bidang Sistem dan Manajemen Basis Data, Pemrograman *Web*. Untuk komunikasi, penulis dapat dihubungi melalui surel: farhanramadhana97@gmail.com.